



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Πληροφοριακό σύστημα για τη διασφάλιση της φαρμακευτικής
συμμόρφωσης χρόνιων ασθενών με τη χρήση ανασχεδιασμένων
φαρμακευτικών συσκευασιών και έξυπνων συσκευών (smart-blister)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Ε. Αντωνιουδάκης

Επιβλέπων : Παναγιώτης Δ. Τσανάκας

Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2025



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Πληροφοριακό σύστημα για τη διασφάλιση της φαρμακευτικής
συμμόρφωσης χρόνιων ασθενών με τη χρήση ανασχεδιασμένων
φαρμακευτικών συσκευασιών και έξυπνων συσκευών (smart-bliстер)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Ε. Αντωνιουδάκης

Επιβλέπων : Παναγιώτης Δ. Τσανάκας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11^η Μαρτίου 2025.

.....
Παναγιώτης Δ. Τσανάκας
Καθηγητής Ε.Μ.Π.

.....
Δημήτριος – Διονύσιος Γ. Κουτσοϋρης
Ομότιμος Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Κ. Ματσόπουλος
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2025

.....

Θεόδωρος Ε. Αντωνιουδάκης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Θεόδωρος Ε. Αντωνιουδάκης, 2025

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Αντικείμενο της διπλωματικής εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου πληροφοριακού συστήματος το οποίο θα παρακολουθεί την φαρμακευτική συμμόρφωση των ασθενών με απώτερο στόχο την βελτίωση ή την διατήρησή της σε υψηλά επίπεδα.

Το πρόβλημα της μη φαρμακευτικής συμμόρφωσης των ασθενών, δηλαδή της μη τήρησης του φαρμακευτικού σχήματος σύμφωνα με τις υποδείξεις των θεραπόντων ιατρών, είναι ένα πολύ σημαντικό πρόβλημα το οποίο δυσχεραίνει την αποκατάσταση της υγείας των ασθενών ή ακόμη χειρότερα εγκυμονεί σοβαρούς κινδύνους επιδείνωσης της ασθένειας ενώ παράλληλα δεν αποκομίζονται ασφαλή συμπεράσματα για την αποτελεσματικότητα των φαρμάκων.

Η εργασία επικεντρώνεται στην ανάπτυξη ενός ολοκληρωμένου πληροφοριακού συστήματος το οποίο καταγράφει τις κινήσεις του ασθενούς όπου σχετίζονται με την φαρμακευτική του αγωγή, τον ειδοποιεί σχετικά με τις μελλοντικές του δόσεις ενώ παράλληλα είναι στη διάθεση αυτού και του θεράποντος ιατρού το πλήρες ιστορικό των κινήσεων του ασθενούς σε σχέση με τη φαρμακευτική του συμμόρφωση.

Πιο συγκεκριμένα το σύστημα αυτό είναι μια υπηρεσία σύννεφου η οποία είναι σε θέση να επικοινωνεί με οποιαδήποτε IoT έξυπνη συσκευή (smart blister) μέσω μιας διεπαφής προγράμματος εφαρμογής (API) με χρήση της αρχιτεκτονικής Representational State Transfer (REST). Το πλεονέκτημα αυτής της αρχιτεκτονικής είναι η απλότητα της καθώς η απόκριση των αιτημάτων γίνεται με μηνύματα μορφής κειμένου HTML ή JSON τα οποία μπορεί να είναι κρυπτογραφημένα αν το πρωτόκολλο είναι ασφαλές HTTP Secure.

Λέξεις κλειδιά

φαρμακευτική συμμόρφωση, ολοκληρωμένο πληροφοριακό σύστημα, έξυπνη συσκευή, smart blister, υπηρεσίες σύννεφου, Διαδίκτυο των Πραγμάτων – IoT, Διεπαφή Προγράμματος Εφαρμογής API

Abstract

The objective of this thesis is the development of an integrated information system that monitors patients' medical adherence, with the ultimate goal of improving or maintaining it at high levels.

The issue of non-adherence to medication, that is the failure of following prescribed treatment regimens according to the instructions of the treating doctor, is a significant problem. It hinders the recovery of the patient, or even worse, poses serious risks of disease progression, while at the same time prevents the extraction of reliable conclusions regarding medical effectiveness.

This research focuses on developing an integrated information system that records the patients' actions, related to their medical adherence, notifies them about the upcoming doses, and provides both patients and doctors a complete history of the patients' medical adherence.

More specifically, this system is a cloud service capable of communicating with any IoT smart device (smart blister) through an Application Program Interface (API), using the Representational State Transfer (REST) architecture. The advantage of this architecture lies in its simplicity, as the response to requests are sent in text format such as HTML or JSON, which can be encrypted if the protocol is HTTP Secure (HTTPS).

Keywords

medical adherence, integrated information system, smart device, smart blister, cloud services, Internet of Things - IoT, Application Program Interface API

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Καθηγητή και Κοσμήτορα της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών κ. Παναγιώτη Τσανάκα για την ευκαιρία που μου έδωσε μέσω αυτής της διπλωματικής να εφαρμόσω τις γνώσεις που απέκτησα κατά την διάρκεια των σπουδών μου σε έναν κλάδο που μέχρι τώρα δεν είχα ασχοληθεί και είναι ο κλάδος της υγείας. Μέσω αυτής της διπλωματικής είναι κατανοητό πως ο ρόλος του μηχανικού είναι πολύ σημαντικός στην εφαρμογή νέων τεχνολογιών στην υπηρεσία του ανθρώπου και γενικότερα της κοινωνίας.

Επίσης θα ήθελα να ευχαριστήσω τους Δρ. Δημήτρη Νικητόπουλο και Δρ. Σόλωνα Ζάννο για την καθοδήγηση και την άμεση και αποτελεσματική ανταπόκριση τους σε θέματα που προέκυπταν κατά την διάρκεια της εκπόνησης της διπλωματικής εργασίας.

Τέλος θα ήθελα να ευχαριστήσω την γυναίκα μου Ρωξάνη και τις δύο μου κόρες Στεφανία και Μαριτίνα για την υπομονή και την κατανόηση που επέδειξαν από την εισαγωγή μου στη Σχολή μέχρι και την ολοκλήρωση της καθώς ο συνδυασμός επαγγελματικών υποχρεώσεων, οικογενειακών υποχρεώσεων και φοίτησης σε μια τόσο απαιτητική Σχολή ήταν ένα αρκετά δύσκολο εγχείρημα όπου όμως στο τέλος αφήνει ένα θετικό πρόσημο.

Περιεχόμενα

Περίληψη	5
Abstract.....	6
Ευχαριστίες	7
Περιεχόμενα.....	8
1. Εισαγωγή.....	13
1.1 Το Αντικείμενο της διπλωματικής.....	13
1.2 Τα οφέλη	13
1.3 Οργάνωση του τόμου.....	14
2. Η φαρμακευτική συμμόρφωση και η σημασία της στη θεραπεία των ασθενών	15
2.1. Ορισμός.....	15
2.2. Στατιστικά στοιχεία	15
2.3. Παράγοντες που επηρεάζουν την φαρμακευτική συμμόρφωση	16
2.3.1. Κοινωνικοί και οικονομικοί παράγοντες.....	16
2.3.2. Παράγοντες που αφορούν την νόσο.....	16
2.3.3. Παράμετροι της θεραπείας	16
2.3.4. Παράγοντες που σχετίζονται με το σύστημα Υγείας	17
2.3.5. Παράγοντες που σχετίζονται με τον ίδιο τον ασθενή.....	17
2.4. Συνέπειες μη συμμόρφωσης στη θεραπεία	17
2.3. Μέθοδοι μέτρησης της φαρμακευτικής συνέπειας.....	18
2.3.1. Υποκειμενική μέθοδος μέτρησης της φαρμακευτικής συνέπειας.....	19
2.3.2. Αντικειμενική μέθοδος μέτρησης της φαρμακευτικής συνέπειας	19
3. Παρουσίαση της προτεινόμενης λύσης Smart Blister	22
3.1 Εισαγωγή.....	22
3.2 Τα συστατικά του Ολοκληρωμένου Πληροφοριακού Συστήματος	23
3.3 Η περιγραφή της web εφαρμογής.....	24
3.3.1 Οι εντολές του API	27
3.3.2 Η εφαρμογή του κινητού.....	29
4. Οι υπηρεσίες σύννεφου (cloud services)	31
4.1. Εισαγωγή.....	31
4.2. Πλεονεκτήματα της χρήσης υπηρεσιών Cloud.	31
4.3. Κατηγορίες υπηρεσιών cloud.....	33
4.3.1 IaaS (Infrastructure as a Service).....	33

4.3.2 PaaS (Platform as a Service)	33
4.3.3 SaaS (Software as a Service).....	34
4.4. Μοντέλα με βάση τους αποδέκτες της υπηρεσίας.....	37
5. Διαδίκτυο των πραγμάτων (Internet of Things) και σύγχρονες τεχνολογικές λύσεις που εφαρμόζονται στο χώρο της Υγείας	39
5.1. Ορισμός του Διαδικτύου των πραγμάτων (Internet of Things)	39
5.2 Η Αρχιτεκτονική του IoT.....	40
5.2.1 Η Αρχιτεκτονική των τριών επιπέδων	40
5.2.2 Η Αρχιτεκτονική των πέντε επιπέδων	41
5.3 Αισθητήρες & ενεργοποιητές.....	41
5.3.1 Αισθητήρες κινητών τηλεφώνων.....	42
5.3.2 Ιατρικοί αισθητήρες	45
5.4. Κύρια ζητήματα και προκλήσεις του IoT	45
5.4.1. Ασφάλεια.....	45
5.4.2. Ιδιωτικότητα	46
5.4.3. Διαλειτουργικότητα και Πρότυπα	46
5.4.4. Νομικά, Ρυθμιστικά και Δικαιώματα	46
5.5 Επικοινωνία.....	47
5.5.1 Επικοινωνίες από συσκευή σε συσκευή	47
5.5.2 Επικοινωνίες από συσκευή σε σύννεφο.....	48
5.5.3. Επικοινωνίες συσκευής προς πύλη.....	48
5.5.4 Πρωτόκολλα επικοινωνίας σε IoT συσκευές	49
5.5.4.1. Message Queuing Telemetry Transport (MQTT).....	51
5.5.4.2. Constrained Application Protocol (CoAP).....	52
5.5.4.3. Advanced Message Queuing Protocol (AMQP).....	52
5.5.4.4. Data Distribution Service (DDS).....	52
5.5.4.5. Extensible Messaging and Presence Protocol (XMPP)	53
5.5.4.6 HTTP/HTTPS Protocol.....	53
5.6. Εφαρμογές IoT	54
5.6.1 Μεταποίηση/Βιομηχανία	54
5.6.2. Μεταφορές	55
5.6.3 Ενέργεια	55
5.6.4. Λιανικό εμπόριο	56
5.6.5 Έξυπνες πόλεις	56
5.6.6. Εφοδιαστική Αλυσίδα	56

5.6.7 Γεωργία	57
5.6.8 Υγειονομική περίθαλψη	57
5.6.9 Σύγχρονες τεχνολογικές λύσεις που εφαρμόζονται στον Τομέα της Υγείας.....	57
5.6.9.1 Απομακρυσμένη Παρακολούθηση Ασθενούς.....	58
5.6.9.2 IoT και τηλεϊατρική	62
5.6.9.3 Έξυπνα νοσοκομεία και IoT εξοπλισμός	62
6. Σχεδιασμός του Πληροφοριακού Συστήματος Smart Blister	63
6.1. Μοντέλο Επιχειρησιακού Σεναρίου	63
6.1.1. Καταγραφή Δραστών.....	63
6.1.2. Ορισμός Δραστών.....	64
6.1.3. Περιγραφή Σεναρίων Χρήσης.....	66
6.1.4. Αναλυτική περιγραφή των σεναρίων.....	67
6.1.5. Διάγραμμα περίπτωσης χρήσης (use case diagram)	70
6.1.6. Διαπροσωπίες Χρηστών.....	71
6.1.7. Μοντέλο Οντοτήτων Συσχετίσεων (Entity Relationship Diagram)	77
6.2. Επιλογή Τεχνολογίας	78
7. Επίλογος (Συμπεράσματα και ιδέες για μελλοντικές επεκτάσεις)	80
7.1. Συμπεράσματα.....	80
7.2. Ιδέες για μελλοντικές επεκτάσεις	80
Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής.....	81
Βιβλιογραφία.....	95

Κατάλογος Σχημάτων

Σχήμα 1. Παράγοντες που επηρεάζουν την συμμόρφωση	17
Σχήμα 2. Οφέλη φαρμακευτικής συμμόρφωσης.....	18
Σχήμα 3. Συνέπειες της μη συμμόρφωσης	19
Σχήμα 4. Το παθητικό ηλεκτρονικό κύκλωμα της κυψέλης φαρμάκων	22
Σχήμα 5. Αρχική σελίδα της εφαρμογής	24
Σχήμα 6. Γραφική αποτύπωση κινήσεων	25
Σχήμα 7. Αναλυτική αποτύπωση κινήσεων.....	25
Σχήμα 8. Δυνατότητα εγγραφής με ένα από τα τρία προφίλ.....	26
Σχήμα 9. Αρχιτεκτονική REST	26
Σχήμα 10. Απόκριση από εντολή ~/smartblister/api/service/	27
Σχήμα 11. Απόκριση από εντολή ~/smartblister/api/status/xxx	28
Σχήμα 12. Απόκριση από εντολή ~/smartblister/api/pillremoved/xxx	28
Σχήμα 13. Απόκριση από εντολή ~/smartblister/api/detailedActions/yy	29
Σχήμα 14. Το interface της mobile εφαρμογής.....	30
Σχήμα 15. Λόγοι στροφής επιχειρήσεων στις υπηρεσίες cloud	32
Σχήμα 16. Παραδείγματα ανά κατηγορία υπηρεσιών cloud.....	35
Σχήμα 17. Η δομή των τριών μοντέλων υπηρεσιών cloud, SaaS, PaaS και IaaS.....	36
Σχήμα 18. Τα όρια ευθύνης ανάμεσα στον χρήστη cloud υπηρεσιών και τον πάροχο cloud υπηρεσιών	37
Σχήμα 19. Cloud computing: μοντέλα παράδοσης, μοντέλα ανάπτυξης, καθορισμός χαρακτηριστικών, πόρων και οργάνωση της υποδομής	38
Σχήμα 20. Πρόβλεψη εξάπλωσης IoT συσκευών	40
Σχήμα 21. Η Αρχιτεκτονική των τριών επιπέδων	41
Σχήμα 22. Η Αρχιτεκτονική των πέντε επιπέδων	41
Σχήμα 23. Επιταχυνσιόμετρο.....	42
Σχήμα 24. Αισθητήρας Οθόνης Αφής	44
Σχήμα 25. Έξυπνα ρολόγια και ιχνηλάτες γυμναστικής στα αριστερά. Ενσωματωμένο έμπλαστρο δέρματος στα δεξιά.....	45
Σχήμα 26. Επικοινωνία από συσκευή σε συσκευή	47
Σχήμα 27. Επικοινωνία από συσκευή σε σύννεφο.....	48
Σχήμα 28. Επικοινωνία συσκευής προς πύλη.....	49

Σχήμα 29. Ασύρματα πρωτόκολλα πρόσβασης	50
Σχήμα 30. Πρωτόκολλα εφαρμογών IoT.....	51
Σχήμα 31. Σύγκριση CoAP με MQTT	52
Σχήμα 32. HTTP μέθοδοι	53
Σχήμα 33. Επίπεδα της Αρχιτεκτονικής Τηλεϊατρικής	58
Σχήμα 34. Όργανο συνεχούς παρακολούθησης γλυκόζης	60
Σχήμα 35. Use Case Diagram	70
Σχήμα 36. Διαπροσωπία Φαρμακοποιού - Αναφορές.....	71
Σχήμα 37. Διαπροσωπία Φαρμακοποιού – Αρχικοποίηση Συσκευής.....	72
Σχήμα 38. Διαπροσωπία Φαρμακοποιού – Αναφορές Προφίλ.....	73
Σχήμα 39. Διαπροσωπία Ιατρού - Αναφορές.....	74
Σχήμα 40. Διαπροσωπία Ιατρού – Επίδοση / Δοσολογία.....	75
Σχήμα 41. Διαπροσωπία Ιατρού – Προφίλ.....	76
Σχήμα 42. Μοντέλο Οντοτήτων Συσχετίσεων.....	77
Σχήμα 43. Γραφικός κώδικας mobile εφαρμογής σε περιβάλλον MIT APP inventor.....	94

Κεφάλαιο 1

Εισαγωγή

Το ολοκληρωμένο πληροφοριακό σύστημα παρακολούθησης της φαρμακευτικής συμμόρφωσης του πληθυσμού στοχεύει στην εξάλειψη ενός πολύ μεγάλου προβλήματος στον χώρο της υγείας που έχει εντοπιστεί από την εποχή του Ιπποκράτη και παραμένει δυστυχώς επίκαιρο ακόμη και στις μέρες μας αφού σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας στις ανεπτυγμένες χώρες, το ποσοστό μη-συμμόρφωσης των ασθενών με τη φαρμακευτική αγωγή για τις χρόνιες ασθένειες, κυμαίνεται στο 50% περίπου, ενώ εμφανίζεται ακόμη υψηλότερο στις αναπτυσσόμενες χώρες.

1.1 Το Αντικείμενο της διπλωματικής

Το σύστημα παρακολούθησης των κινήσεων του ασθενούς σχετικά με την φαρμακευτική του αγωγή είναι ένα σύστημα που θα βοηθήσει τον ασθενή στην διατήρηση της φαρμακευτικής του συνέπειας, ενώ παράλληλα ο θεράπωντας ιατρός του θα έχει την πλήρη και ακριβή εικόνα σχετικά με την αγωγή του ασθενούς με αποτέλεσμα να μπορεί να αντλήσει αξιόπιστα συμπεράσματα σχετικά με την επίδραση της φαρμακευτικής αγωγής στην πορεία της υγείας του.

Ο θεράπωντας ιατρός παράλληλα με την χορήγηση της φαρμακευτικής συνταγής θα δίνει ακριβείς οδηγίες σχετικά με αυτήν. Συγκεκριμένα θα δίνει το φαρμακευτικό σχήμα της θεραπείας, δηλαδή την ποσότητα, την συχνότητα και την διάρκεια της θεραπείας μέσω μιας web-based εφαρμογής. Στην συνέχεια ο φαρμακοποιός θα αρχικοποιεί την συσκευή Smart Blister και θα την παραδίδει στον ασθενή.

Η εφαρμογή έχει υλοποιηθεί σε γλώσσα python μέσω του Django Framework, τα δεδομένα αποθηκεύονται σε βάση δεδομένων SQL Light και όλη η υποδομή βρίσκεται σε πάροχο φιλοξενίας εικονικών μηχανών Virtual Machines (VMs). Το Smart Blister θα είναι μία Internet of Things (IoT) συσκευή η οποία μεταδίδει τα δεδομένα μέσω δικτύου στενής ζώνης Narrowband στην εφαρμογή.

1.2 Τα οφέλη

Τα οφέλη του συστήματος αυτού θα είναι άμεσα στην αντιμετώπιση της έλλειψης φαρμακευτικής συμμόρφωσης των ασθενών με σημαντικές επιπτώσεις τόσο στο επίπεδο της υγείας των ασθενών όσο και σε οικονομικό και κοινωνικό επίπεδο. Αναλυτικά τα οφέλη της χρήσης του συστήματος αυτού είναι τα εξής:

- Βελτίωση της ποιότητας ζωής των χρόνιων ασθενών
- Γρηγορότερη ανάρρωση των ασθενών
- Μείωση παρενεργειών από λανθασμένες συνταγογραφήσεις
- Μείωση των νοσοκομειακών νοσηλειών
- Περιορισμός της σπατάλης πόρων στην χορήγηση φαρμάκων

- Ανατροφοδότηση των ιατρών με ακριβείς και έγκυρες πληροφορίες σχετικά με την αποτελεσματικότητα των φαρμάκων.

1.3 Οργάνωση του τόμου

Η δομή της διπλωματικής εργασίας είναι:

Κεφάλαιο 2 Ανάλυση της έννοιας της φαρμακευτικής συμμόρφωσης και των συνεπειών της έλλειψης αυτής. Στατιστικά στοιχεία, οφέλη της φαρμακευτικής συμμόρφωσης, αίτια μη τήρησης αυτής, καθώς και μέθοδοι επίλυσης του προβλήματος.

Κεφάλαιο 3 Παρουσίαση της προτεινόμενης πλατφόρμας με σύντομη αναφορά κάθε υποσυστήματος που την απαρτίζει.

Κεφάλαιο 4 Υπηρεσίες σύννεφου (cloud services). Παρουσιάζονται οι διάφοροι τύποι υπηρεσιών σύννεφου που παρέχονται στους οργανισμούς και τις επιχειρήσεις και τα οφέλη που μπορούν να αποκομίσουν από την χρήση των υπηρεσιών αυτών έναντι της χρήσης υποδομών που υπάρχουν με φυσικό τρόπο στον χώρο τους.

Κεφάλαιο 5 Διαδίκτυο των Πραγμάτων (Internet of Things) και σύγχρονες τεχνολογικές λύσεις που εφαρμόζονται στο χώρο της Υγείας. Αναλύεται η έννοια του Διαδικτύου των Πραγμάτων, ο τρόπος υλοποίησης τους καθώς και παραδείγματα χρήσης τους στην εποχή μας με απώτερο στόχο την βελτίωση των υπηρεσιών που παρέχονται. Ιδιαίτερη αναφορά γίνεται στις λύσεις που έχουν εφαρμοστεί στον χώρο της υγείας.

Κεφάλαιο 6 Αναλυτική παρουσίαση της λύσης που έχει υλοποιηθεί. Παρουσιάζονται αναλυτικά τα εργαλεία που έχουν χρησιμοποιηθεί τόσο σε επίπεδο λογισμικού όσο και σε επίπεδο υλικού.

Κεφάλαιο 7 Συμπεράσματα και ιδέες για επεκτάσεις της λύσης που παρουσιάστηκε.

Κεφάλαιο 8 Ενδεικτικό μέρος του κώδικα της εφαρμογής.

Κεφάλαιο 2

Η φαρμακευτική συμμόρφωση και η σημασία της στη θεραπεία των ασθενών

2.1. Ορισμός

Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας η φαρμακευτική συμμόρφωση αφορά τη λήψη του σωστού φαρμακευτικού σκευάσματος, την κατάλληλη στιγμή, στην απαιτούμενη δόση και για το χρονικό διάστημα που συστήνει ο εκάστοτε θεράπων ιατρός.

Η φαρμακευτική συμμόρφωση του ασθενούς στις οδηγίες του θεράποντος ιατρού είναι ένας πολύ κρίσιμος και καθοριστικός παράγοντας στην βελτίωση της υγείας του ασθενούς και στην ποιότητα ζωής των ανθρώπων που πάσχουν από χρόνιες ασθένειες.

Το πρόβλημα της μειωμένης φαρμακευτικής συμμόρφωσης είχε εντοπιστεί από την εποχή του Ιπποκράτη 2000 χρόνια πριν και είναι ακόμη επίκαιρη και ακόμη πιο έντονη στην εποχή μας. Επίσης ο Ιπποκράτης έλεγε χαρακτηριστικά ότι πριν θεραπεύσεις κάποιον, ρώτησέ τον εάν είναι πρόθυμος να αφήσει όλα εκείνα που τον κάνουν να αρρωσταίνει. Πόσο όμως έντονο είναι το πρόβλημα της μη φαρμακευτικής συμμόρφωσης και γιατί δυσκολεύονται οι ασθενείς να πειθαρχήσουν στις οδηγίες των ιατρών;

2.2. Στατιστικά στοιχεία

Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας στις ανεπτυγμένες χώρες, η συμμόρφωση των ασθενών σε χρόνιες ασθένειες είναι κατά μέσο όρο μόνο 50%. Το ποσοστό αυτό είναι ακόμη μικρότερο στις αναπτυσσόμενες χώρες λόγω της δεδομένης έλλειψης πόρων υγείας και των ανισοτήτων πρόσβασης στην υγειονομική περίθαλψη. [1]

Για παράδειγμα, στην Κίνα, στη Γκάμπια και τις Σεϋχέλλες, μόνο το 43%, το 27% και το 26%, αντίστοιχα, των ασθενών με υπέρταση τηρούν το φαρμακευτικό σχήμα τους. Στις αναπτυγμένες χώρες όπως οι Ηνωμένες Πολιτείες, μόνο το 51% των ασθενών που υποβάλλονται σε θεραπεία για υπέρταση τηρούν την συνταγογραφούμενη θεραπεία. Τα δεδομένα για ασθενείς με κατάθλιψη αποκαλύπτουν ότι το ποσοστό της φαρμακευτικής συμμόρφωσης κυμαίνεται μεταξύ 40% και 70% στις αντικαταθλιπτικές θεραπείες. Στην Αυστραλία, μόνο το 43% των ασθενών με άσθμα λαμβάνουν τα φάρμακά τους όπως ακριβώς συνταγογραφούνται και μόνο το 28% χρησιμοποιεί συνταγογραφούμενα προληπτικά φάρμακα. Στη θεραπεία του HIV και AIDS, η συμμόρφωση κυμαίνεται μεταξύ 37% και 83% ανάλογα με το φάρμακο και τα δημογραφικά χαρακτηριστικά των πληθυσμών. Αυτό αντιπροσωπεύει μια τεράστια θλιβερή πρόκληση για τις προσπάθειες διατήρησης της υγείας του πληθυσμού, όπου η επιτυχία καθορίζεται κυρίως από την τήρηση των μακροχρόνιων θεραπειών.

Τέλος η συμμόρφωση όχι μόνο στην λήψη κάποιας φαρμακευτικής αγωγής αλλά στον συνδυασμό φαρμακευτικής αγωγής με την αλλαγή στον τρόπο ζωής κυμαίνεται πολύ χαμηλότερα μεταξύ 20%-30%

2.3. Παράγοντες που επηρεάζουν την φαρμακευτική συμμόρφωση

Το πρόβλημα είναι αρκετά σύνθετο και δύσκολο στην αντιμετώπισή του δεδομένου ότι υπάρχουν αρκετοί παράγοντες που πρέπει να αντιμετωπιστούν και πολλοί από αυτούς δυστυχώς δεν εξαρτώνται από τον ίδιο τον ασθενή.

2.3.1. Κοινωνικοί και οικονομικοί παράγοντες

Το εισόδημα, η κοινωνική θέση, η εκπαίδευση καθώς και η γεωγραφική θέση του ασθενούς είναι μερικοί από τους παράγοντες αυτής της κατηγορίας. Όταν ο ασθενής βρίσκεται σε μια απομακρυσμένη περιοχή όπου δεν είναι εφικτή η άμεση επαφή του με τον θεράποντα ιατρό απαιτείται επιπλέον προσπάθεια όπου έχει κόστος σε χρόνο και σε χρήμα. Το πρόβλημα γίνεται ακόμη πιο έντονο όταν ο ασθενής είναι σε προχωρημένη ηλικία και η μετακίνηση του καθίσταται ιδιαίτερα δύσκολη.

2.3.2. Παράγοντες που αφορούν την νόσο

Η σοβαρότητα, ο ρυθμός εξέλιξης και τα συμπτώματα της νόσου μπορούν να οδηγήσουν τον ασθενή σε μεγαλύτερη συνέπεια. Αντιθέτως αν τα συμπτώματα δεν είναι άμεσα αλλά μπορεί να εμφανιστούν μετά από μεγάλο χρονικό διάστημα δημιουργούν μια χαλάρωση στην ασθενή καθώς εφησυχάζεται και πιστεύει ότι δεν θα εξελιχθεί η ασθένεια απειλητικά εναντίων του. Σε αυτό το σημείο μεγάλο ρόλο θα διαδραματίσει η πειθώ του θεράποντος ιατρού για τον κίνδυνο που θα αντιμετωπίσει ο ασθενής στο μέλλον αν δεν είναι συνεπής και το μορφωτικό επίπεδο του ασθενούς ώστε να κατανοήσει και να λάβει σοβαρά υπόψιν του τα λεγόμενα του ιατρού. Τέλος η διάρκεια της νόσου είναι ένας ακόμη παράγοντας που επηρεάζει την συμμόρφωση των ασθενών. Συγκεκριμένα όπως είπαμε και πριν το ποσοστό συμμόρφωσης για μακροχρόνιες ασθένειες ανέρχεται στο 40%-50% ενώ στις βραχυχρόνιες ασθένειες το ποσοστό ανεβαίνει στο 70%-80% καθώς η κόπωση είναι ένας ανασταλτικός παράγοντας στην πειθαρχία των ασθενών.

2.3.3. Παράμετροι της θεραπείας

Η περιπλοκότητα του θεραπευτικού σχήματος, η μη αμεσότητα των αποτελεσμάτων της θεραπείας ή και η πιθανή εμφάνιση κάποιων παρενεργειών των φαρμάκων αποτρέπει τον ασθενή στην τήρηση της θεραπείας όπως έχει οριστεί από τον θεράποντα ιατρό. Εδώ ο ρόλος του ιατρού είναι καθοριστικός καθώς θα πρέπει να δώσει όσο το δυνατό πιο απλό θεραπευτικό σχήμα στο μέτρο πάντα του εφικτού. Επίσης θα πρέπει να είναι πειστικός όταν η θεραπεία δεν έχει άμεσα αποτελέσματα αλλά πρόκειται να αποδώσει μακροπρόθεσμα. Τέλος θα πρέπει ο ιατρός να έχει επαφή με τον ασθενή και να εξετάζει κατά πόσο η θεραπεία έχει ομαλή εξέλιξη και δεν προκαλεί κάποια παρενέργεια. Στην αντίθετη περίπτωση θα πρέπει να είναι σε θέση να αλλάξει το θεραπευτικό σχήμα όπου θα βοηθήσει στην αποφυγή παρενεργειών χωρίς να χάνει σε αποτελεσματικότητα. Φυσικά η στενή επαφή ιατρού και ασθενή είναι πολύ σημαντική στις περιπτώσεις των ανθρώπων όπου πάσχουν με χρόνιες ασθένειες κάτι που όμως δεν είναι πάντα εφικτό γιατί από την μια ο ιατρός παρακολουθεί πάρα πολλούς ασθενείς κυρίως σε δημόσιες

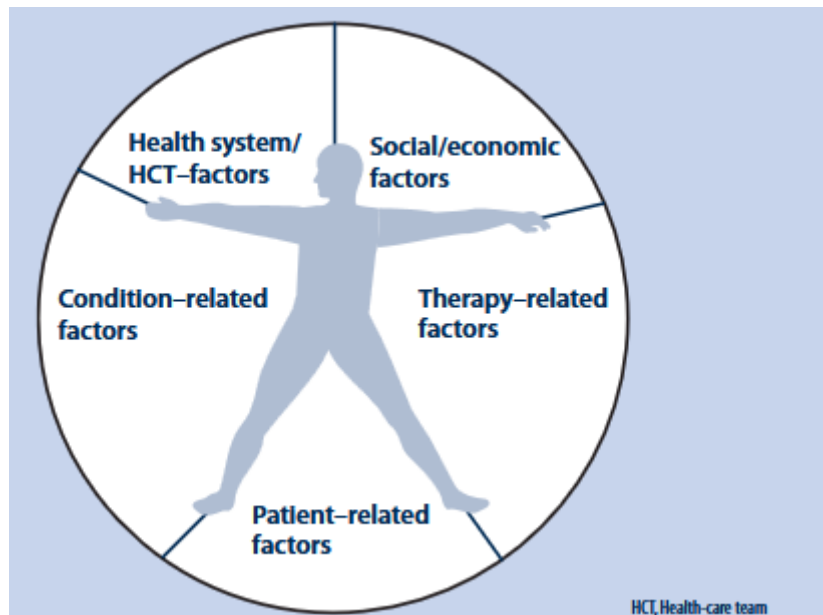
δομές και από την άλλη πολλές φορές ο ασθενής δεν διαθέτει τους πόρους, δηλαδή τον χρόνο και το χρήμα για τακτικές επισκέψεις.

2.3.4. Παράγοντες που σχετίζονται με το σύστημα Υγείας

Η ποιότητα της σχέσης ασθενούς-ιατρού, η ευκολία πρόσβασης στο Σύστημα Υγείας και οι επιλογές της θεραπείας είναι παράγοντες που μπορούν να βοηθήσουν τον ασθενή κατά την θεραπείας του. Είναι πολύ σημαντικό να υπάρχει εύκολη πρόσβαση του ασθενούς στο ΣΥ τόσο για να προσεγγίσει τον θεράποντα ιατρό όσο και για να μπορεί να κάνει τις προβλεπόμενες εξετάσεις προκειμένου να έχει αυτός και ο ιατρός του πλήρη και ξεκάθαρη εικόνα για την κατάσταση της υγείας του. Επίσης οι εναλλακτικές επιλογές θεραπείας μπορούν να βοηθήσουν τον ιατρό στην εύρεση μιας θεραπείας που θα ταιριάζει στον εκάστοτε ασθενή ανάλογα με τις ανάγκες που τις δυσκολίες που μπορεί να αντιμετωπίζει.

2.3.5. Παράγοντες που σχετίζονται με τον ίδιο τον ασθενή

Ένα μεγάλο πλήθος ασθενών δεν διαθέτει υγειονομική παιδεία, γεγονός που θέτει σε κίνδυνο την υγεία τους. Επίσης εμπειρίες των ασθενών στο παρελθόν με παρόμοιες θεραπείες, η αρνητική σκέψη ως προς την εξέλιξη της νόσου, η κακή ψυχολογία που μπορεί να οφείλετε και σε άλλα προβλήματα όπου πιθανόν αντιμετωπίζει παράλληλα ο ασθενής καθιστούν δύσκολη την όλη διαδικασία της θεραπείας. [2]



Σχήμα 1. Παράγοντες που επηρεάζουν την συμμόρφωση

2.4. Συνέπειες μη συμμόρφωσης στη θεραπεία

Το μεγάλο ποσοστό της φαρμακευτικής συμμόρφωσης βοηθάει τους ασθενείς να απολαύσουν τα μέγιστα κλινικά και οικονομικά οφέλη από τη θεραπευτική τους αγωγή καθώς βελτιώνεται η ποιότητα ζωής τους, βελτιώνεται το επίπεδο της υγείας τους και αποτρέπεται ο πρόωρος

θάνατος ενώ παράλληλα μειώνεται και το κόστος υγειονομικής τους περίθαλψης. Αντίθετα το μικρό ποσοστό φαρμακευτικής συνέπειας δεν αποδίδει τα επιθυμητά επίπεδα ως προς την εξέλιξη της υγείας του ασθενούς αλλά αντίθετα επιδεινώνει την ασθένεια με αποτέλεσμα την χορήγηση περισσότερων φαρμάκων και την πιθανή εισαγωγή του ασθενούς σε κάποια μονάδα υγείας. Αποτέλεσμα αυτού είναι ο ασθενής να γίνεται περισσότερο ασυνεπής στην φαρμακευτική αγωγή και έτσι δημιουργείται ένας φαύλος κύκλος. Γενικότερα η μη φαρμακευτική συμμόρφωση έχει ολέθριες συνέπειες στην υγεία του ασθενούς και επιβαρύνει οικονομικά τον ίδιο αλλά και γενικότερα το Σύστημα Υγείας.



Σχήμα 2. Οφέλη φαρμακευτικής συμμόρφωσης [3]

2.3. Μέθοδοι μέτρησης της φαρμακευτικής συνέπειας

Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας η φαρμακευτική συνέπεια είναι άρρηκτα συνδεδεμένη με τη λήψη του σωστού φαρμακευτικού σκευάσματος, την κατάλληλη στιγμή, στην απαιτούμενη δόση και για το χρονικό διάστημα που συστήνει ο εκάστοτε θεράπων ιατρός. Συνεπώς μια μετρική που θα μπορούσαμε να χρησιμοποιήσουμε είναι ο λόγος του πλήθους των χαπιών που έχει καταναλώσει ο ασθενής προς το συνολικό πλήθος χαπιών που έχει υποδείξει στην εκάστοτε θεραπεία ο θεράπων ιατρός. Με ποιους τρόπους όμως μπορεί να καταγραφεί αυτή η μετρική; Υπάρχουν δύο μέθοδοι καταγραφής της φαρμακευτικής συνέπειας. Η υποκειμενική μέθοδος και η αντικειμενική μέθοδος. [4] [5]



Σχήμα 3. Συνέπειες της μη συμμόρφωσης [3]

2.3.1. Υποκειμενική μέθοδος μέτρησης της φαρμακευτικής συνέπειας

Κατά την υποκειμενική μέθοδο μέτρησης της φαρμακευτικής συνέπειας ο θεράπων ιατρός κάνει μια μικρή συζήτηση με τον ασθενή, παράλληλα και με τους εγγυτέρους συγγενείς του σχετικά με την συνέπειά του στην τήρηση της θεραπευτικής αγωγής. Πιθανότατα η συζήτηση αυτή θα μπορούσε να συνοδεύεται και με κάποιο ερωτηματολόγιο. [6] Στην περίπτωση που η αγωγή έχει ένα σχετικά μεγάλο βάθος χρόνου ο ασθενής δεν έχει πάντα την ακριβή εικόνα ως προς την συνέπεια του ή ακόμα χειρότερα δεν θέλει να δώσει την πλήρη εικόνα αν γνωρίζει ότι δεν ήταν απόλυτα συνεπής. Το αποτέλεσμα είναι να μην μπορεί να βγάλει ο ιατρός τα ακριβή συμπεράσματα για το κατά πόσο η θεραπεία λειτουργεί ανασταλτικά στην εξέλιξη της νόσου. Αν η συζήτηση αυτή συνοδεύεται και από κάποιες εξετάσεις που δεν έχουν τα επιθυμητά αποτελέσματα είναι πολύ πιθανό να προχωρήσει ο ιατρός σε πιο επιθετικό σχήμα που θα μπορούσε ή ακόμη καλύτερα θα έπρεπε να είχε αποφευχθεί. Γενικότερα η υποκειμενική μέθοδος έχει αποδειχθεί ότι είναι μια αναξιόπιστη μέθοδος και μπορεί να φέρει τα εντελώς αντίθετα από τα επιθυμητά αποτελέσματα.

2.3.2. Αντικειμενική μέθοδος μέτρησης της φαρμακευτικής συνέπειας

Η πιο απλή αντικειμενική μέθοδος καταγραφής της φαρμακευτικής συνέπειας είναι αυτή κατά την οποία ο ασθενής προσκομίζει τα υπόλοιπα φάρμακα από το σύνολο των φαρμάκων που του χορηγήθηκαν. Αν για παράδειγμα χορηγήθηκαν στον ασθενή 60 χάπια με συχνότητα 2 ανά ημέρα και μετά από ένα μήνα θεραπείας ο ασθενής επέστρεψε 12 χάπια ο ιατρός θα συμπεράνει ότι ο ασθενής είχε φαρμακευτική συνέπεια $(60-12)/60 \times 100 = 80\%$. Αυτή είναι μια ακριβής μέτρηση ως προς την ποσότητα των φαρμάκων που έλαβε ο ασθενής σε σχέση με αυτή που θα

έπρεπε να λάβει, όμως δεν μπορούν να βγουν συμπεράσματα σχετικά με τον ακριβή χρόνο λήψης των φαρμάκων αυτών. Αν δηλαδή διατηρήθηκαν τα επιθυμητά χρονικά διαστήματα ανάμεσα σε δύο διαδοχικές δόσεις φαρμάκων. [4]

Μια ακόμη αντικειμενική μέθοδος είναι αυτή κατά την οποία μέσα στο φάρμακο προστίθενται μη τοξικοί βιολογικοί δείκτες οι οποίοι μπορούν να εντοπιστούν με ειδικές εξετάσεις στο αίμα ή στα ούρα. [7] Αυτή η μέθοδος δεν είναι απόλυτα ακριβής καθώς ο βαθμός εντοπισμού των δεικτών αυτών εξαρτάται από τον μεταβολισμό του ασθενούς και από την διατροφή του. Επίσης αυτή η μέθοδος εντοπίζει την σχετικά πρόσφατη χορήγηση των φαρμάκων και όχι την γενική συνέπεια του ασθενούς από την αρχή της θεραπείας του. Επίσης παρατηρείται αρκετά συχνά το φαινόμενο να είναι περισσότερο συνεπής ο ασθενής λίγες μέρες πριν τις προγραμματισμένες εξετάσεις του. Αυτό δυσκολεύει ακόμη περισσότερο την προσπάθεια καταγραφής της φαρμακευτικής συνέπειας του ασθενούς.

Επίσης η καταγραφή και ανάλυση των δεδομένων του συνόλου των συνταγογραφήσεων και των πωλήσεων των αντίστοιχων φαρμάκων είναι και αυτή μια ακόμη αντικειμενική μέθοδος. Προϋποθέτει την καταγραφή των δεδομένων από τα δύο ανεξάρτητα συστήματα, του συστήματος συνταγογραφήσεων των ιατρών και του συστήματος πωλήσεων των φαρμάκων από τα φαρμακεία και την αποτύπωση τους σε ένα ενοποιημένο σύστημα ώστε να γίνει κατόπιν η διασταύρωση των στοιχείων. [7] Αυτή όμως είναι μια μέθοδος εξαιρετικά δύσκολη δεδομένου οι ιατροί δεν είναι απαραίτητο να γράφουν συγκεκριμένο φάρμακο αλλά τη δραστική ουσία που περιέχει το φάρμακο η οποία θα πρέπει να συσχετιστεί με ένα μεγάλο σύνολο γενόσημων φαρμάκων. Παράλληλα με αυτόν τον τρόπο δεν μπορούμε να βγάλουμε συμπεράσματα για την φαρμακευτική συμμόρφωση ενός συγκεκριμένου ασθενή αλλά για το σύνολο του ευρύτερου πληθυσμού. Επίσης και στην περίπτωση που θέλουμε να βγάλουμε συμπεράσματα για την φαρμακευτική συμμόρφωση του ευρύτερου πληθυσμού τα συμπεράσματα που θα βγουν δεν είναι ασφαλή καθώς η συμμόρφωση δεν συσχετίζεται αποκλειστικά και μόνο με την αγορά των φαρμάκων.

Ένας ακόμη τρόπος αντικειμενικής καταγραφής της συνέπειας του ασθενούς είναι η χρήση ειδικών συσκευασιών φαρμάκων στα οποία υπάρχει ενσωματωμένο ηλεκτρονικό κύκλωμα στην συσκευή φυσαλίδων φαρμάκων (blister). Με το κύκλωμα αυτό καταγράφεται η ακριβής ώρα αφαίρεσης του κάθε χαπιού και αποθηκεύεται στην μνήμη. Μετά αυτή η συσκευή μπορεί να συνδεθεί με κάποιον υπολογιστή και από εκεί θα μπορέσουμε να συλλέξουμε τα καταγεγραμμένα δεδομένα. Αυτή η μέθοδος χαρακτηρίζεται από μεγάλη ακρίβεια έχει όμως δύο σημαντικά μειονεκτήματα. [8] Τον αυξημένο όγκο της συσκευασίας φαρμάκου και το υψηλό κόστος κατασκευής του που είναι απαγορευτικό για μαζική παραγωγή. Έτσι αυτές οι συσκευές μπορούν να χρησιμοποιηθούν από ένα καλό στατιστικό δείγμα μόνο για να δώσουν στις φαρμακευτικές εταιρείες δεδομένα για την επίδοση του φαρμάκου και όχι για την καταγραφή της φαρμακευτικής συνέπειας του ευρύτερου πληθυσμού.

Μία περίπτωση που συνδυάζει την ακρίβεια των αποτελεσμάτων καταγραφής με το χαμηλό κόστος κατασκευής είναι η χρήση ενός παθητικού κυκλώματος στο blister των φαρμάκων στο οποίο θα υπάρχουν αντιστάσεις εκτυπωμένες με ειδικά αγώγιμα μελάνια. Η συνολική αντίσταση του blister μεταβάλλεται καθώς αφαιρούνται τα χάπια αφού κόπτονται οι σχετικές αντιστάσεις που βρίσκονται στο σημείο αφαίρεσης του χαπιού. Το blister η καλύτερα οι αντιστάσεις του blister διαβάζονται από ειδική συσκευή στην οποία συνδέεται το blister και στέλνουν την πληροφορία αυτή σε ένα εξωτερικό web service με το οποίο επικοινωνεί η συσκευή αυτή. Το πλεονέκτημα αυτής της λύσης είναι ότι το κόστος της δημιουργίας των blister είναι αρκετά χαμηλό με αποτέλεσμα να είναι οικονομικά εφικτή η μαζική παραγωγή του. Η ειδική συσκευή είναι επαναχρησιμοποιούμενη καθώς μπορεί να ενσωματωθεί με κάθε

Η φαρμακευτική συμμόρφωση και η σημασία της στη θεραπεία των ασθενών

νέο blister στο οποίο θέλουμε να γίνει η καταγραφή των κινήσεων του ασθενούς, ενώ παράλληλα το blister είναι πρακτικά στο ίδιο μέγεθος με το κλασικό blister.

Κεφάλαιο 3

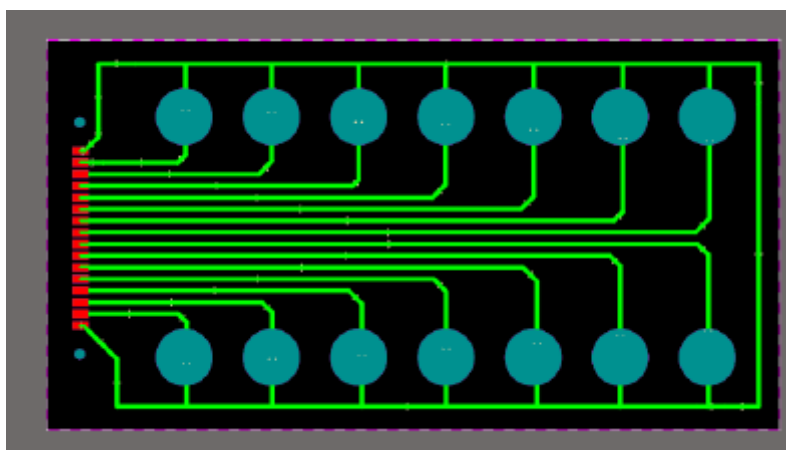
Παρουσίαση της προτεινόμενης λύσης Smart Blister

3.1 Εισαγωγή

Όπως είδαμε στο προηγούμενο κεφάλαιο υπάρχουν δύο κύριοι αντικειμενικοί και αξιόπιστοι τρόποι καταγραφής της φαρμακευτικής συμμόρφωσης των ασθενών. Και οι δύο απαιτούν ειδικές συσκευές καταγραφής της αφαίρεσης των χαπιών από τις ειδικές κυψέλες που περιέχουν τα χάπια.

Στην πρώτη περίπτωση υπάρχει ένα ολοκληρωμένο κύκλωμα με μνήμη και ρολόι και ώστε να μπορούμε να διαβάσουμε τις κινήσεις μέσω μιας δεύτερης συσκευής ανάγνωσης των κινήσεων. Χρησιμοποιεί τεχνολογία Radio Frequency Identifier RFID. Το μειονέκτημα αυτής της λύσης είναι το υψηλό κόστος παραγωγής και ότι η μετάδοση των δεδομένων δεν γίνεται σε πραγματικό χρόνο, αλλά ασύγχρονα με την τοποθέτηση της συσκευής ανάγνωσης. Η λύση αυτή δεν προσφέρεται για μαζική παραγωγή λόγω του υψηλού κόστους όπως προείπαμε, θα μπορούσε όμως να χρησιμοποιηθεί από τις φαρμακευτικές εταιρείες για να βγάλουν ασφαλή συμπεράσματα για την αποτελεσματικότητα ενός φαρμάκου που έχουν προωθήσει στην αγορά. Στην δεύτερη περίπτωση χρησιμοποιείται ένα σύστημα το οποίο αποτελείται από (1) μια έξυπνη κυψέλη αποθήκευσης φαρμάκων, η οποία είναι ένα παθητικό ηλεκτρονικό κύκλωμα, (2) μια μικρή ηλεκτρονική συσκευή, ο ενεργοποιητής, ο οποίος διαβάσει τις αντιστάσεις τις έξυπνης κυψέλης φαρμάκων και (3) ένα πληροφοριακό σύστημα το οποίο λαμβάνει μηνύματα από τον ενεργοποιητή.

Η έξυπνη κυψέλη περιέχει ειδικά αγωγίμα μελάνια που διασχίζουν τα σημεία των κυψελών κάτω από τα οποία βρίσκονται τα χάπια. Όταν αφαιρεθεί ένα χάπι αλλάζει η τιμή της αντίστασης της κυψέλης και ο ενεργοποιητής στέλνει το μήνυμα στο πληροφοριακό σύστημα σχετικά με το χάπι που αφαιρέθηκε καθώς και για τον ακριβή χρόνο αυτής της κίνησης. [9]



Σχήμα 4. Το παθητικό ηλεκτρονικό κύκλωμα της κυψέλης φαρμάκων

Η επικοινωνία του ενεργοποιητή με το πληροφοριακό σύστημα μπορεί να γίνει με τρεις διαφορετικούς τρόπους:

- NB-IoT (Narrowband IoT) δίκτυο
- Bluetooth Low Energy (BLE) με ενδιάμεση κινητή συσκευή η οποία λαμβάνει το μήνυμα και το προωθεί στο δίκτυο μέσω εφαρμογής που βρίσκεται στην κινητή συσκευή.
- WiFi κατά την οποία δεν χρειάζεται ενδιάμεση κινητή συσκευή αλλά επικοινωνεί απ' ευθείας ο ενεργοποιητής με το δίκτυο και κατά συνέπεια με το πληροφοριακό σύστημα. Η διαφορά σε σχέση με την πρώτη λύση είναι ότι ο ασθενής θα πρέπει να βρίσκεται σε κάποιο δίκτυο WiFi ενώ παράλληλα υπάρχει μεγαλύτερη κατανάλωση ενέργειας από την συσκευή.

Τα πλεονεκτήματα της λύσης αυτής σε σχέση με αυτή της πρώτης περίπτωσης είναι ότι ο ενεργοποιητής μπορεί να χρησιμοποιηθεί σε κάθε καινούργια κυψέλη που θα χορηγηθεί στον ασθενή. Παραμένει δηλαδή η ηλεκτρονική συσκευή του ενεργοποιητή στην κατοχή του ασθενή για μελλοντική χρήση, ενώ είναι εφικτή η μαζική παραγωγή των κυψελών των χαπιών με το παθητικό κύκλωμα καθώς το κόστος παραγωγής τους είναι πολύ χαμηλό.

Σαν συνέχεια της παραπάνω λύσης θα υλοποιήσουμε ένα ολοκληρωμένο πληροφοριακό σύστημα το οποίο θα αποτελείται από μια web based εφαρμογή και θα δέχεται μηνύματα από οποιαδήποτε ηλεκτρονική συσκευή είναι σε θέση να στέλνει μηνύματα σχετικά με τις κινήσεις του ασθενούς ως προς τη λήψη φαρμάκων. Συνεπώς η συσκευή θα είναι μια οποιαδήποτε ηλεκτρονική συσκευή η οποία θα μπορεί να διαβάζει ένα παθητικό κύκλωμα με αντιστάσεις όπου θα είναι τυπωμένο στην κυψέλη (blister) και θα στέλνει μηνύματα με τις κατάλληλες προδιαγραφές στο web service. Ποιο συγκεκριμένα το service αυτό θα στέλνει μηνύματα που βασίζονται στην τεχνολογία Representational State Transfer (REST)

3.2 Τα συστατικά του Ολοκληρωμένου Πληροφοριακού Συστήματος

Το σύστημα που θα δημιουργήσουμε αποτελείται από τα κάτωθι συστατικά:

- Μία εικονική μηχανή, virtual machine, στην οποία έχει εγκατεσταθεί λειτουργικό σύστημα ubuntu έκδοσης 18.04.6 LTS (Codename bionic)
- Ένας εξυπηρετητής διαδικτύου (web server) Apache 2.4.29
- Ένας εξυπηρετητής βάσεων δεδομένων (database server) SQL Lite
- Ένα framework για αποδοτική παραγωγή web εφαρμογής σε γλώσσα python – Django Framework 3.0.7
- Ένα front-end Framework, το bootstrap έκδοσης 4.0, για την αποδοτική δημιουργία ιστοσελίδων. Χρησιμοποιείται για τη δημιουργία ιστοσελίδων όπου προσαρμόζονται αυτόματα σε διαφορετικά μεγέθη οθονών (desktop, laptop, tablet, laptop). Παρέχει έτοιμα στυλ, grid system και JavaScript components, διευκολύνοντας την ανάπτυξη μοντέρνων ιστοσελίδων χωρίς να χρειάζεται να γράψουμε όλο τον CSS και JavaScript κώδικα από το μηδέν.
- Μία web based εφαρμογή στην οποία εμπεριέχεται και το REST api το οποίο μπορεί να επικοινωνήσει με οποιαδήποτε συσκευή η οποία μπορεί να στείλει REST μηνύματα.
- Μία εφαρμογή κινητού η οποία υλοποιήθηκε για να στέλνει REST μηνύματα στην web εφαρμογή και θα λειτουργήσει ως προσομοιωτής της ηλεκτρονικής συσκευής smart blister. Η εφαρμογή αναπτύχθηκε στο περιβάλλον MIT app inventor.

Η εικονική μηχανή φιλοξενείται σε πάροχο υπηρεσιών διαδικτύου (cloud services). Η υπηρεσία που χρησιμοποιήσαμε από τον πάροχο είναι τύπου Infrastructure as a Service IaaS

καθώς ο πάροχος μας έδωσε την εικονική μηχανή και εμείς κάναμε όλες τις απαιτούμενες εγκαταστάσεις και δημιουργήσαμε το απαιτούμενο λογισμικό για επικοινωνία με τις έξυπνες συσκευές. Πάνω σε αυτή την μηχανή έχουν εγκατασταθεί ο web και ο database server. Το Django Framework δεν έχει εγκατασταθεί απ' ευθείας στην εικονική μηχανή αλλά σε εικονικό περιβάλλον, virtual environment. Η χρήση το εικονικού περιβάλλοντος βοηθάει στον να απομονωθούν οι εξαρτήσεις που μπορεί να έχουν τα περιφερειακά υποπρογράμματα το συστήματος με το έργο και αυτό θα βοηθήσει στο να είναι η εγκατάσταση ανεξάρτητη της εκάστοτε πλατφόρμας που μπορεί να το φιλοξενήσει. Έτσι διασφαλίζεται η φορητότητα της εγκατάστασης σε οποιοδήποτε νέο περιβάλλον με ιδιαίτερη ευκολία.

3.3 Η περιγραφή της web εφαρμογής

Η εφαρμογή χρησιμοποιείται από τους ασθενείς, τους ιατρούς και τους φαρμακοποιούς. Αρχικά ο ιατρός δημιουργεί μια συνταγή στην οποία περιγράφει την ασθένεια και καταχωρεί το φάρμακο που πρέπει να χορηγηθεί και την δοσολογία του φαρμάκου που πρέπει να χορηγηθεί στον ασθενή. Ουσιαστικά καταγράφει το θεραπευτικό σχήμα που θα χορηγηθεί στον ασθενή.

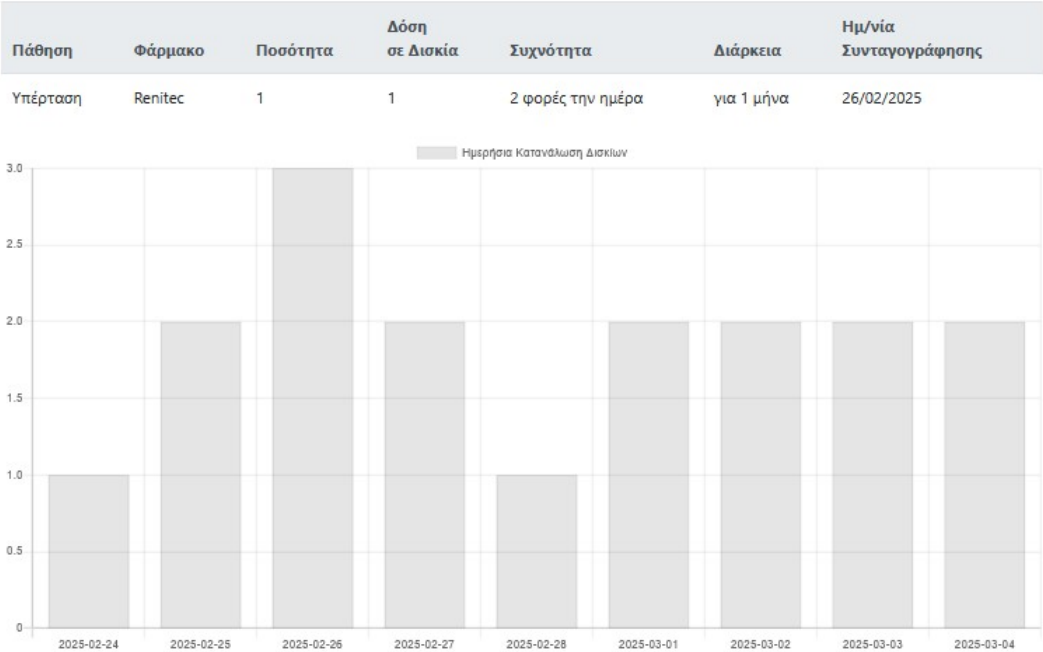


Σχήμα 5. Αρχική σελίδα της εφαρμογής

Κατόπιν ο φαρμακοποιός εντοπίζει τον ασθενή από το σύστημα, του χρεώνει την συσκευή smart blister την οποία και αρχικοποιεί συνδέοντας την με την συνταγή που χορηγήθηκε από τον ιατρό. Η συσκευή παραλαμβάνεται από τον ασθενή και μέσω αυτής καταγράφονται στο σύστημα οι κινήσεις του, σχετικά με την λήψη του φαρμάκου. Ο θεράπων ιατρός θα μπορέσει να κάνει χρήση των στοιχείων που έχουν καταγραφεί στο σύστημα αφού πρώτα πάρει την σχετική έγκριση από τον ασθενή. Το αίτημα για χρήση των στοιχείων και η έγκριση αυτής γίνεται πάλι μέσω του συστήματος αυτού. Συγκεκριμένα όταν αιτηθεί ο ιατρός πρόσβαση στις κινήσεις του ασθενούς αποστέλλεται ένα ηλεκτρονικό μήνυμα σε αυτόν και μετά την αποδοχή του αιτήματος από τον ασθενή λαμβάνει κανονικά τα στοιχεία των κινήσεων του σε σχέση με την χορήγηση των φαρμάκων. Οι πληροφορίες εμφανίζονται σε μορφή γραφήματος για να έχει μια καλύτερη εποπτική εικόνα αλλά και οι λεπτομέρειες των κινήσεων που θα βοηθήσουν το ιατρό να καταλάβει όχι απλά αν ο ασθενής πήρε τα φάρμακά του αλλά και αν τα πήρε τις σωστές χρονικές στιγμές.

Παρουσίαση της προτεινόμενης λύσης Smart Blister

Φαρμακευτική Συνέπεια Ασθενούς



Σχήμα 6. Γραφική αποτύπωση κινήσεων

Αναλυτικές Κινήσεις στο Blister	
A/A	Ημ/νία Αφαίρεσης
1	04/03/2025 10:47 μμ.
2	04/03/2025 10:47 πμ.
3	03/03/2025 10:36 μμ.
4	03/03/2025 10:36 πμ.
5	02/03/2025 10:35 μμ.
6	02/03/2025 10:35 πμ.
7	01/03/2025 10:34 πμ.
8	01/03/2025 12:10 πμ.
9	28/02/2025 10:34 πμ.
10	27/02/2025 10:33 μμ.
11	27/02/2025 10:28 πμ.
12	26/02/2025 10:21 μμ.
13	26/02/2025 10:21 πμ.
14	26/02/2025 2:45 πμ.
15	25/02/2025 10:27 μμ.
16	25/02/2025 10:26 πμ.
17	24/02/2025 10:28 μμ.

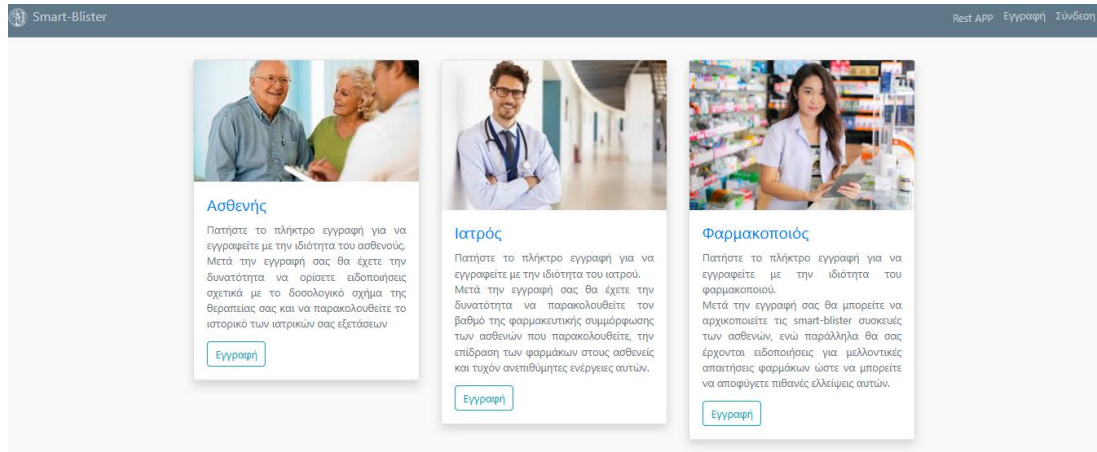
Επιστροφή

Σχήμα 7. Αναλυτική αποτύπωση κινήσεων

Παρουσίαση της προτεινόμενης λύσης Smart Blister

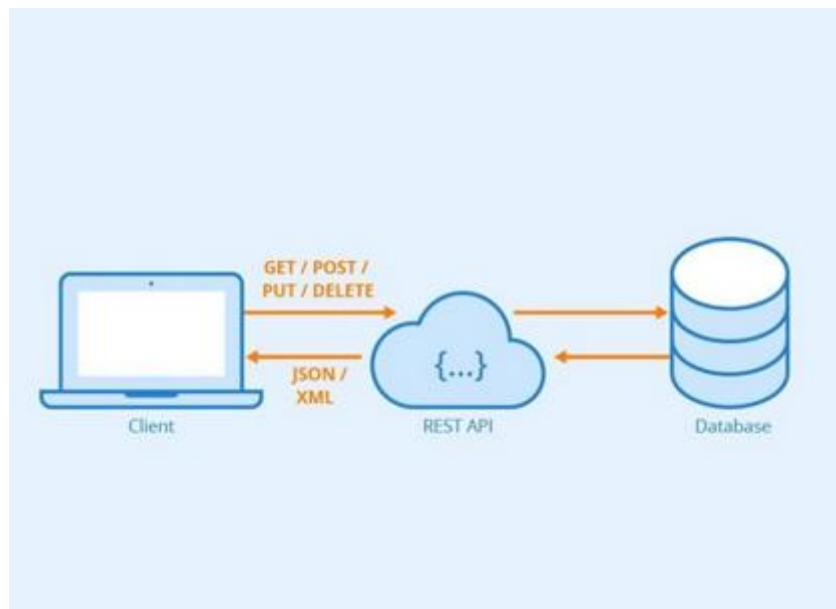
Σχήμα 3.3. Αναλυτική αποτύπωση κινήσεων

Φυσικά δίνεται η δυνατότητα στις τρεις βασικές οντότητες, ασθενής, ιατρός, φαρμακοποιός να κάνουν εγγραφή στο σύστημα, ενώ παράλληλα μπορούν ανά πάσα στιγμή να κάνουν ενημέρωση των προσωπικών τους στοιχείων.



Σχήμα 8. Δυνατότητα εγγραφής με ένα από τα τρία προφίλ

Στην εφαρμογή έχει ενσωματωθεί και ένα REST API το οποίο είναι μια διεπαφή, για την διασύνδεση του συστήματος αυτού με την εκάστοτε συσκευή smart blister. Το REST api είναι μια συνήθης αρχιτεκτονική που χρησιμοποιείται για να μπορούν διαφορετικά συστήματα να επικοινωνούν μεταξύ τους με έναν αρκετά απλό και αξιόπιστο τρόπο. Ουσιαστικά ανταλλάσσουν αρχεία κειμένου με συγκεκριμένη μορφή, κυρίως JSON ή XML και συγκεκριμένη δομή και σύνταξη που ορίζεται εκ των προτέρων από τον κατασκευαστή του Application Program Interface API.



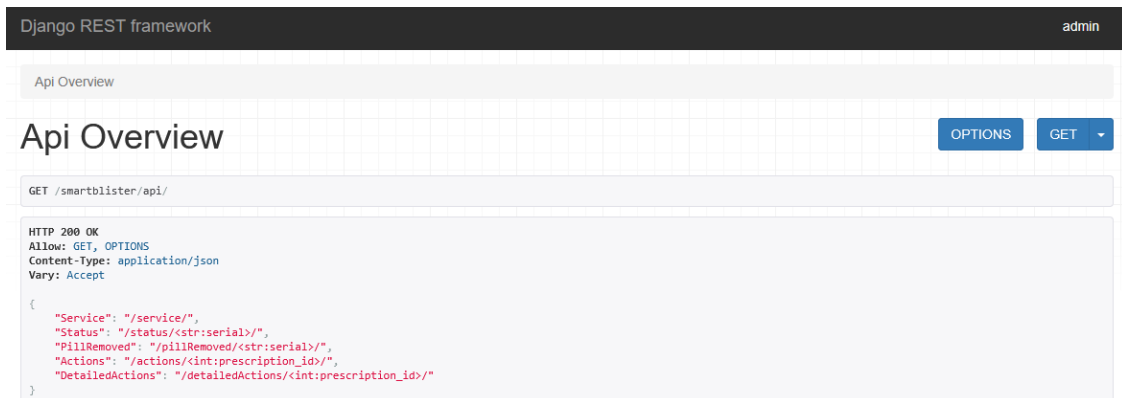
Σχήμα 9. Αρχιτεκτονική REST

3.3.1 Οι εντολές του API

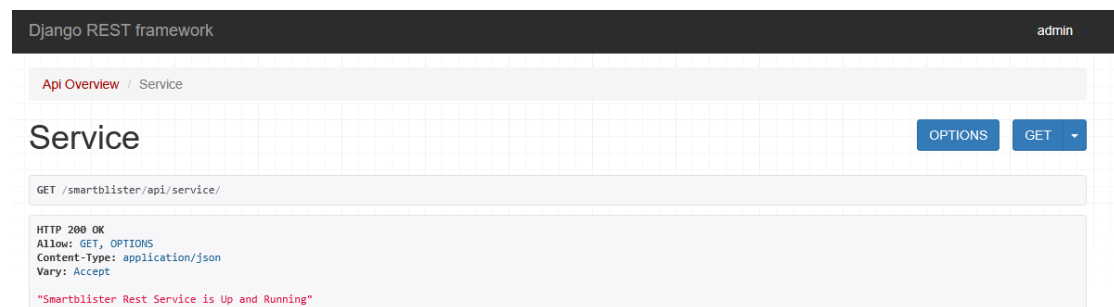
Οι εντολές που μπορεί να λάβει το API είναι η παρακάτω:

- `~/smartblister/api/` Σε αυτή την εντολή απαντάει το σύστημα με το σύνολο των εντολών που είναι διαθέσιμες από το API
- `~/smartblister/api/service/` Στην περίπτωση που το σύστημα είναι σε λειτουργία ενημερώνει για την ενεργή του κατάσταση
- `~/smartblister/api/status/xxx` Όπου xxx είναι ο σειριακός αριθμός της smartblister συσκευής. Με αυτή την εντολή δίνονται πληροφορίες που σχετίζονται με το συγκεκριμένο blister. Συγκεκριμένα δίνεται η ημερομηνία χρέωσης του smartblister, ο ασθενής στον οποίο χρεώθηκε η συσκευή, η πάθηση του, ο ιατρός του, το φάρμακό που του συνταγογραφήθηκε καθώς και οι λεπτομέρειες σχετικά με τη δοσολογία του φαρμάκου.
- `~/smartblister/api/pillremoved/xxx` Όπου xxx είναι ο σειριακός αριθμός της smartblister συσκευής. Δηλώνεται η αφαίρεση ενός χαπιού από το blister
- `~/smartblister/api/actions/yy` Όπου yy είναι ο κωδικός της συνταγής. Εδώ εμφανίζονται οι συνολικές ημερήσιες κινήσεις αφαίρεσης χαπιών
- `~/smartblister/api/detailedActions/yy` Όπου yy είναι ο κωδικός της συνταγής. Εμφανίζονται αναλυτικά οι χρονικές στιγμές αφαίρεσης των χαπιών

Στα παρακάτω σχήματα εμφανίζονται ενδεικτικά κάποιες απαντήσεις του REST API σε αντίστοιχα requests



Σχήμα 3.6. Απόκριση από εντολή `~/smartblister/api/`



Σχήμα 10. Απόκριση από εντολή `~/smartblister/api/service/`

Παρουσίαση της προτεινόμενης λύσης Smart Blister

Django REST framework admin

Api Overview / Status

Status

OPTIONS GET

GET /smartblister/api/status/123/

```
HTTP 200 OK
Allow: GET, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "blister": {
    "serial": "123",
    "charge_date": "2025-02-26",
    "patient": {
      "user": {
        "last_name": "Σταυρόπουλος",
        "first_name": "Νικόλαος"
      },
      "doctor": {
        "id": 5,
        "user": {
          "last_name": "Θεοδωρίδης",
          "first_name": "Γεώργιος"
        },
        "speciality": "Παθολόγος",
        "office_place": "Ηλιούπολη"
      }
    },
    "prescription": {
      "doctor": {
        "id": 5,
        "user": {
          "last_name": "Θεοδωρίδης",
          "first_name": "Γεώργιος"
        },
        "speciality": "Παθολόγος",
        "office_place": "Ηλιούπολη"
      },
      "illness": "Υπέρταση",
      "medicine": "Renitec"
    }
  }
}
```

Σχήμα 11. Απόκριση από εντολή ~/smartblister/api/status/xxx

Django REST framework admin

Api Overview / Pill Removed

Pill Removed

OPTIONS

POST /smartblister/api/pillRemoved/123/

```
HTTP 200 OK
Allow: OPTIONS, POST
Content-Type: application/json
Vary: Accept

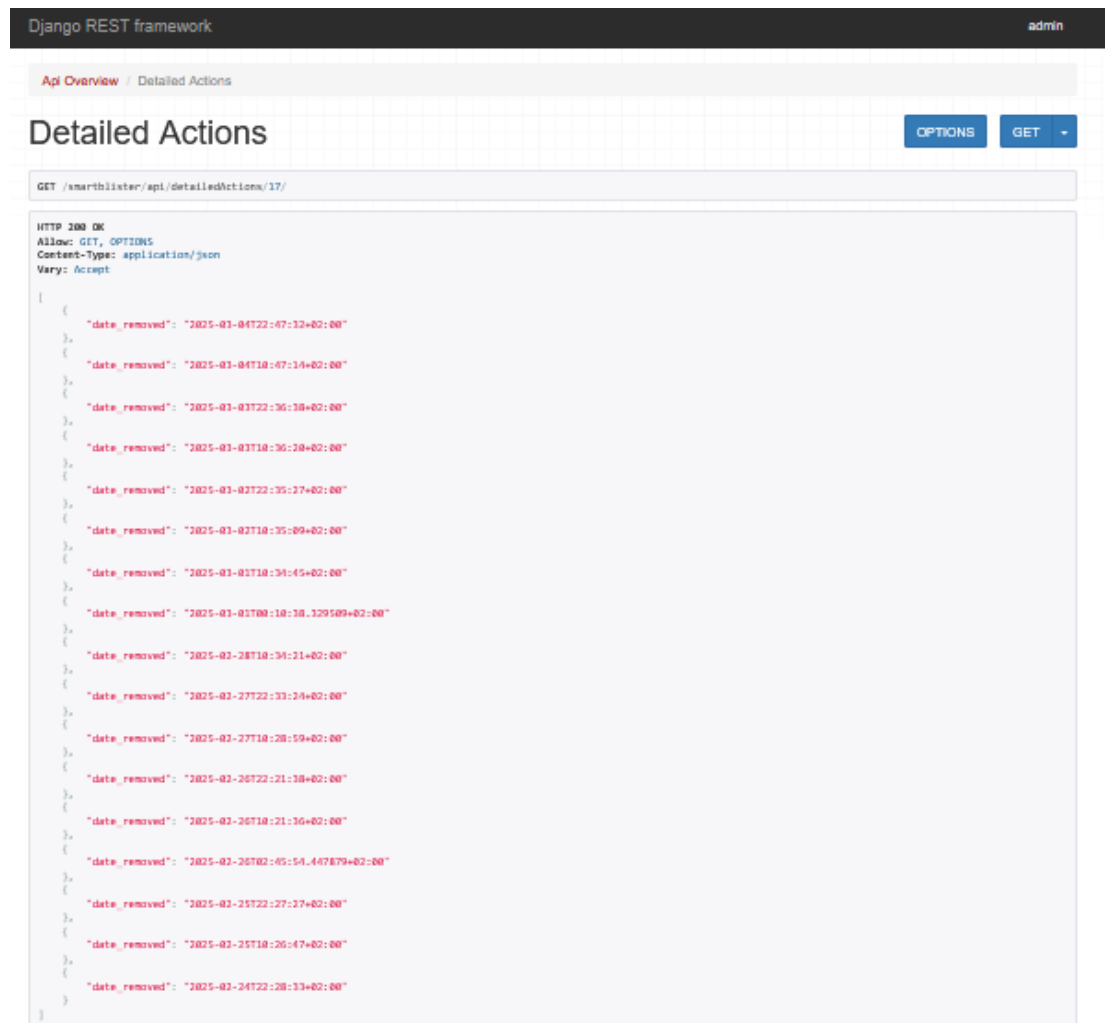
"Η αφαίρεση διακίου καταχωρήθηκε"
```

Media type:

Content:

POST

Σχήμα 12. Απόκριση από εντολή ~/smartblister/api/pillremoved/xxx



Σχήμα 13. Απόκριση από εντολή ~/smartblister/api/detailedActions/yy

3.3.2 Η εφαρμογή του κινητού

Η εφαρμογή του κινητού υλοποιήθηκε μέσω της πλατφόρμας MIT App Inventor η οποία είναι μια οπτική, blocked based πλατφόρμα όπου επιτρέπει στους χρήστες να δημιουργήσουν εφαρμογές κινητού του λειτουργικού συστήματος Android. Σχεδιάστηκε στο Πανεπιστήμιο της Μασαχουσέτης Massachusetts Institute of Technology MIT με έμφαση στη δημιουργία εφαρμογών σε οπτικό και πολύ εύχρηστο περιβάλλον χωρίς να απαιτούνται ιδιαίτερες γνώσεις προγραμματισμού κινητών συσκευών και ο δημιουργός της εφαρμογής μπορεί να αξιοποιήσει τους πόρους του κινητού, όπως κάμερα, ημερολόγιο, αισθητήρες, χάρτες στην υλοποίηση της εφαρμογής με πολύ απλό και αποδοτικό τρόπο. Παράλληλα δίνεται η δυνατότητα αποθήκευσης της εφαρμογής στο cloud.

Η εφαρμογή που υλοποιήθηκε κάνει κλήσεις στο REST API όπως ακριβώς θα έκανε και μια smart blister συσκευή. Μπορούμε για παράδειγμα να κάνουμε ένα request αφαίρεσης χαπιού από το blister.

Παρουσίαση της προτεινόμενης λύσης Smart Blister



Σχήμα 14. Το interface της mobile εφαρμογής

Κεφάλαιο 4

Οι υπηρεσίες σύννεφου (cloud services)

4.1. Εισαγωγή Οι υπηρεσίες cloud είναι τεχνολογικές λύσεις που επιτρέπουν την αποθήκευση, επεξεργασία και διαχείριση δεδομένων μέσω του διαδικτύου, έναντι των τοπικών διακομιστών ή των προσωπικών υπολογιστών. Οι υπηρεσίες αυτές έχουν αναπτυχθεί ραγδαία τα τελευταία χρόνια λόγω πολλών τεχνολογικών, επιχειρηματικών και κοινωνικών παραγόντων. Ο λόγος της ανάπτυξης των υπηρεσιών cloud είναι η βελτίωση των υποδομών σε πολλά επίπεδα. Η εξέλιξη του διαδικτύου, τα ισχυρά data centers οι τεχνολογίες εικονοποίησης (virtualization) και η ανάπτυξη του υπολογιστικού νέφους (cloud computing) με τη χρήση καταναμεμημένων υπολογιστικών πόρων οδήγησε αναπόφευκτα στην εξέλιξη των υπηρεσιών αυτών και την υιοθέτηση της χρήσης αυτών των υπηρεσιών από πολλές επιχειρήσεις και Οργανισμούς.

4.2. Πλεονεκτήματα της χρήσης υπηρεσιών Cloud.

Οι επιχειρήσεις όπου υιοθετούν cloud υπηρεσίες αποκομίζουν από αυτή τη διαδικασία τεράστια οικονομικά οφέλη. Το κόστος της διατήρησης φυσικών servers σε τοπικό επίπεδο απαιτεί ειδικούς χώρους φιλοξενίας αυτών των μηχανημάτων σε χαμηλές θερμοκρασίες με αδιάλειπτη λειτουργία ηλεκτρικής παροχής και παράλληλα θα πρέπει μέσα σε αυτά τα κόστη να υπολογιστεί και το προσωπικό που οφείλει να συντηρεί αυτά τα μηχανήματα. Το προσωπικό αυτό θα πρέπει να είναι ένα προσωπικό υψηλών προσόντων όπου θα έχει την ευθύνη της ασφάλειας των δεδομένων, της ενημέρωσης της υποδομής του λογισμικού με τις πιο πρόσφατες εκδόσεις, της συντήρησης του υλικού και του δικτύου κ.α. Στο κόστος φυσικά θα πρέπει να υπολογίσουμε και την αγορά αυτών των μηχανημάτων όπου έχουν ένα περιορισμένο κύκλο ζωής.

Παράλληλα οι υπηρεσίες αυτές βοηθούν στην ασφάλεια και την εγκυρότητα των δεδομένων. Οι cloud providers επενδύουν σε τεχνολογίες ασφαλείας για την προστασία των δεδομένων. Επίσης κανονιστικές ρυθμίσεις και νόμοι όπως το General Data Protection Regulation GDPR ενθαρρύνουν τη χρήση ασφαλών και συμμορφωμένων cloud υπηρεσιών.

Ένα άλλο σημαντικό πλεονέκτημα που αποκομίζουν οι επιχειρήσεις από τη χρήση αυτών των υπηρεσιών είναι η ευελιξία και η κλιμάκωση. Μπορούν να ξεκινήσουν με μια μικρή υποδομή και ανάλογα με την εξέλιξη της επιχείρησης και του όγκου των δεδομένων που μπορεί να αποκτήσει να μεγαλώσει την υποδομή της με μεγάλη ευκολία. Στην περίπτωση του φυσικού μηχανήματος η επιχείρηση θα έπρεπε να προβεί στην αντικατάσταση των μηχανημάτων με άλλα μεγαλύτερης ισχύος και χωρητικότητας. Επίσης οι επιχειρήσεις δεν έχουν σταθερό φόρτο εργασίας για όλο το χρονικό διάστημα αλλά παρατηρούνται αυξομειώσεις ανάλογα με την περίοδο και την φύση της επιχείρησης. Για παράδειγμα ένα ταξιδιωτικό πρακτορείο θα μπορούσε να έχει μεγαλύτερη επισκεψιμότητα λίγο πριν την περίοδο των διακοπών και παράλληλα αυτή η επισκεψιμότητα να είχε αισθητή μείωση μια άλλη τυχαία χρονική περίοδο. Στην περίπτωση αυτή η εταιρεία θα έπρεπε να δώσει προδιαγραφές για φυσικό μηχάνημα και ταχύτητα διαδικτύου με βάση τις μέγιστες απαιτήσεις που θα μπορούσαν να προκύψουν ενώ

σε μια υπηρεσία cloud είναι εφικτή η αυξομείωση των απαιτούμενων υποδομών και παράλληλα θα ήταν εφικτή η κλιμακωτή χρέωση της υπηρεσίας.

Η τηλεργασία και η απομακρυσμένη πρόσβαση δεν θα μπορούσε να είναι εφικτή σε τόσο μεγάλο βαθμό χωρίς την χρήση cloud-based εργαλείων. Φυσικά μπορούμε και πάλι να εντοπίσουμε εύκολα τα οφέλη τέτοια εργαλείων από τη μεριά και των επιχειρήσεων αλλά και από την μεριά των εργαζομένων. Δεν είναι λίγες οι επιχειρήσεις έχουν στις τάξεις τους προσωπικό όπου εργάζεται από οποιοδήποτε μέρος του πλανήτη με μεγάλη ευκολία. Αυτό δίνει ένα μεγάλο όπλο για τους υπαλλήλους οι οποίοι ξεπερνούν έτσι το εμπόδιο της απόστασης του τόπου διαμονής τους από την έδρα της εταιρείας, αυξάνει όμως έτσι και τον ανταγωνισμό στην αγορά εργασίας. Παράλληλα οι εταιρείες μειώνουν τα κτιριακά και λειτουργικά κόστη αφού μειώνοντας το προσωπικό που εργάζεται στην φυσική έδρα της εταιρείας μειώνουν τα μεγέθη των κτιρίων και τις ανάγκες για θέρμανση, φωτισμό, τηλεφωνία, διαδικτύου, εξοπλισμό γραφείου κ.λ.π.

Ένας ακόμη κλάδος όπου αναπτύσσεται με την χρήση των cloud υπηρεσιών είναι το AI σε συνδυασμό με τα big data. Άλλωστε η συγκέντρωση μεγάλου όγκου δεδομένων είναι απαραίτητο συστατικό της τεχνολογίας τεχνητής νοημοσύνης AI. Η άμεση και εύκολη διασύνδεση εφαρμογών με συστήματα αποθήκευσης δεδομένων μέσω διαδικτύου βοηθάει καταλυτικά στην ανάπτυξη τέτοιων τεχνολογιών.

Η συνεχής πρόοδος της τεχνολογίας και η αυξανόμενη ανάγκη για ευέλικτες, ασφαλείς και οικονομικές λύσεις καθιστούν τις cloud υπηρεσίες αναπόσπαστο μέρος της σύγχρονης ψηφιακής εποχής.

The reasons driving the decision to use public clouds.	
Reason	Percentage who agree
Improved system reliability and availability	50%
Pay only for what you use	50%
Hardware savings	47%
Software license saving	46%
Lower labor costs	44%
Lower maintenance costs	42%
Reduced IT support needs	40%
Ability to take advantage of the latest functionality	40%
Less pressure on internal resources	39%
Solve problems related to updating/upgrading	39%
Rapid deployment	39%
Ability to scale up resources to meet the needs	39%
Ability to focus on core competencies	38%
Take advantage of the improved economics of scale	37%
Reduced infrastructure management needs	37%
Lower energy costs	29%
Reduced space requirements	26%
Create new revenue streams	23%

Σχήμα 15. Λόγοι στροφής επιχειρήσεων στις υπηρεσίες cloud [10]

4.3. Κατηγορίες υπηρεσιών cloud

Υπάρχουν διαφορετικά επίπεδα υπηρεσιών cloud ανάλογα με τις ανάγκες που απαιτούνται από την εκάστοτε εταιρεία και με το κατά πόσο ενεργά θα μπορεί ή θα θελήσει να συμμετάσχει στις παρεχόμενες υπηρεσίες που θα προσφέρει η εταιρεία. Όπως είπαμε και πριν ο βαθμός συμμετοχής της επιχείρησης στην παροχή των προσφερόμενων υπηρεσιών είναι ανάλογος του κόστους των υποδομών καθώς και του προσωπικού σε πλήθος και προσόντα.

4.3.1 IaaS (Infrastructure as a Service)

Παρέχει υποδομές πληροφορικής, όπως εικονικούς διακομιστές, αποθηκευτικό χώρο και δίκτυα. Είναι ουσιαστικά ένα μοντέλο υπολογιστικού νέφους (Cloud Computing) που παρέχει εικονικοποιημένες υποδομές πληροφορικής μέσω διαδικτύου. Αντί να αγοράζονται και να συντηρούνται φυσικοί διακομιστές (servers), αποθηκευτικά μέσα και δικτυακός εξοπλισμός, μια επιχείρηση μπορεί να νοικιάζει υποδομές από έναν Cloud Provider και να τις διαχειρίζεται μέσω διαδικτυακής διεπαφής. [11]

Χαρακτηριστικά του IaaS

- **Υποδομή κατ' απαίτηση** – Οι επιχειρήσεις μπορούν να προσθέτουν ή να αφαιρούν πόρους (CPU, RAM, storage) ανάλογα με τις ανάγκες τους.
- **Εικονικοποίηση** – Οι φυσικοί διακομιστές μετατρέπονται σε **εικονικές μηχανές (VMs)**, διαμοιράζοντας τους πόρους δυναμικά.
- **Ευέλικτη χρέωση (Pay-as-you-go)** – Πληρώνεις μόνο για τους πόρους που χρησιμοποιείς.
- **Ασφάλεια και συντήρηση** – Οι πάροχοι διαχειρίζονται τη φυσική ασφάλεια, τις αναβαθμίσεις και τις ενημερώσεις των κέντρων δεδομένων.
- **Διαχείριση μέσω διεπαφών διαδικτύου** – Παρέχεται έλεγχος και αυτοματοποίηση υποδομών μέσω διεπαφών προγραμματισμού.

Μερικά παραδείγματα IaaS υπηρεσιών είναι η παροχή υπολογιστικής ισχύς (Compute), η αποθήκευση (Storage), η δικτύωση (Networking) και η ασφάλεια (Security)

Παραδείγματα: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP).

4.3.2 PaaS (Platform as a Service)

Το PaaS είναι ένα μοντέλο Cloud Computing που παρέχει μια πλήρη πλατφόρμα ανάπτυξης και διαχείρισης εφαρμογών μέσω διαδικτύου. Σε αντίθεση με το IaaS, όπου οι χρήστες διαχειρίζονται μόνοι τους τις υποδομές (servers, networking, storage), το PaaS παρέχει ένα έτοιμο περιβάλλον για ανάπτυξη, δοκιμή και εκτέλεση εφαρμογών χωρίς να απαιτείται η διαχείριση των υποδομών. [11]

Χαρακτηριστικά του PaaS

- **Έτοιμο περιβάλλον ανάπτυξης** – Περιλαμβάνει όλα τα εργαλεία για ανάπτυξη εφαρμογών (databases, runtime, middleware).
- **Αυτοματοποιημένη διαχείριση υποδομών** – Ο πάροχος αναλαμβάνει τη συντήρηση των εξυπηρετητών, των λειτουργικών συστημάτων και συστημάτων ασφάλειας.
- **Επεκτασιμότητα (Scalability)** – Αυτόματη κλιμάκωση εφαρμογών ανάλογα με τη ζήτηση.
- **Υποστήριξη πολλών γλωσσών & frameworks** – Node.js, Python, Java, PHP, .NET, Ruby, κ.ά.
- **Συνεργατική ανάπτυξη** – Δυνατότητα εργασίας από διαφορετικές τοποθεσίες με κοινά εργαλεία.

Παραδείγματα: Google App Engine, Microsoft Azure App Services, Heroku.

4.3.3 SaaS (Software as a Service)

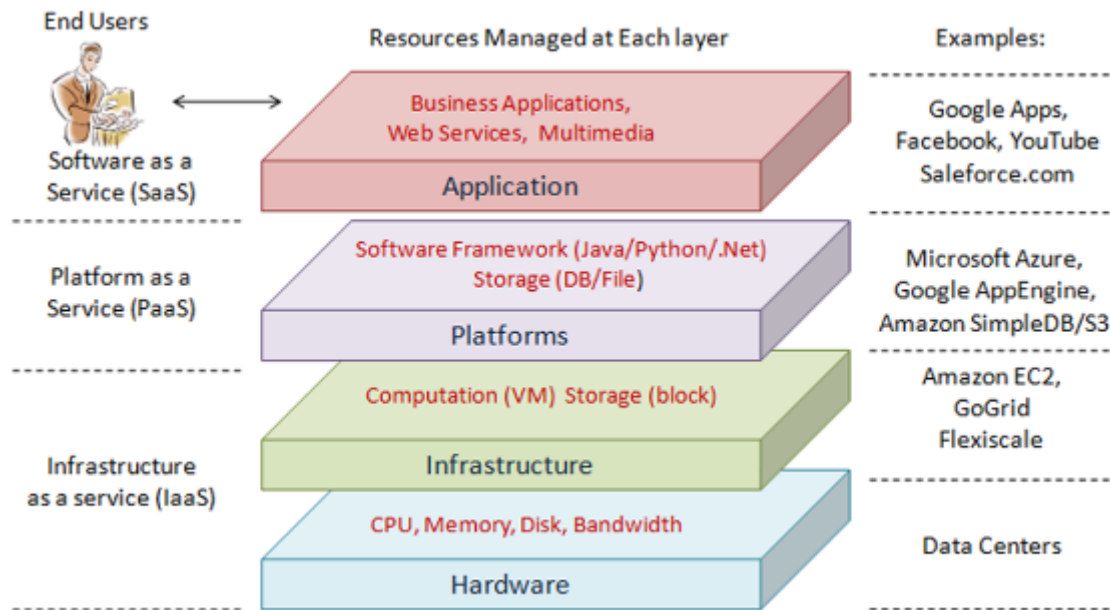
Το SaaS είναι ένα μοντέλο Cloud Computing που επιτρέπει στους χρήστες να εκτελούν εφαρμογές μέσω διαδικτύου χωρίς να χρειάζεται εγκατάσταση ή διαχείριση του λογισμικού. Ο πάροχος SaaS αναλαμβάνει τη φιλοξενία, συντήρηση, αναβαθμίσεις και ασφάλεια της εφαρμογής, ενώ ο χρήστης έχει πρόσβαση σε αυτή μέσω ενός web browser.

Χαρακτηριστικά του SaaS

- **Δεν απαιτεί εγκατάσταση** – Όλα γίνονται μέσω του διαδικτύου.
- **Αυτόματες ενημερώσεις & συντήρηση** – Ο πάροχος διαχειρίζεται τις αναβαθμίσεις και την ασφάλεια.
- **Ευκολία πρόσβασης** – Δυνατότητα χρήσης από οποιαδήποτε συσκευή με σύνδεση στο διαδίκτυο.
- **Συνδρομητικό μοντέλο (Pay-as-you-go)** – Οι χρήστες πληρώνουν ανάλογα με τη χρήση τους.
- **Ενσωματωμένη επεκτασιμότητα** – Μπορεί να προσαρμόζεται σε μεγάλες ή μικρές επιχειρήσεις.
- **Υποστήριξη συνεργασίας** – Πολλαπλοί χρήστες μπορούν να συνεργάζονται ταυτόχρονα σε ένα περιβάλλον.

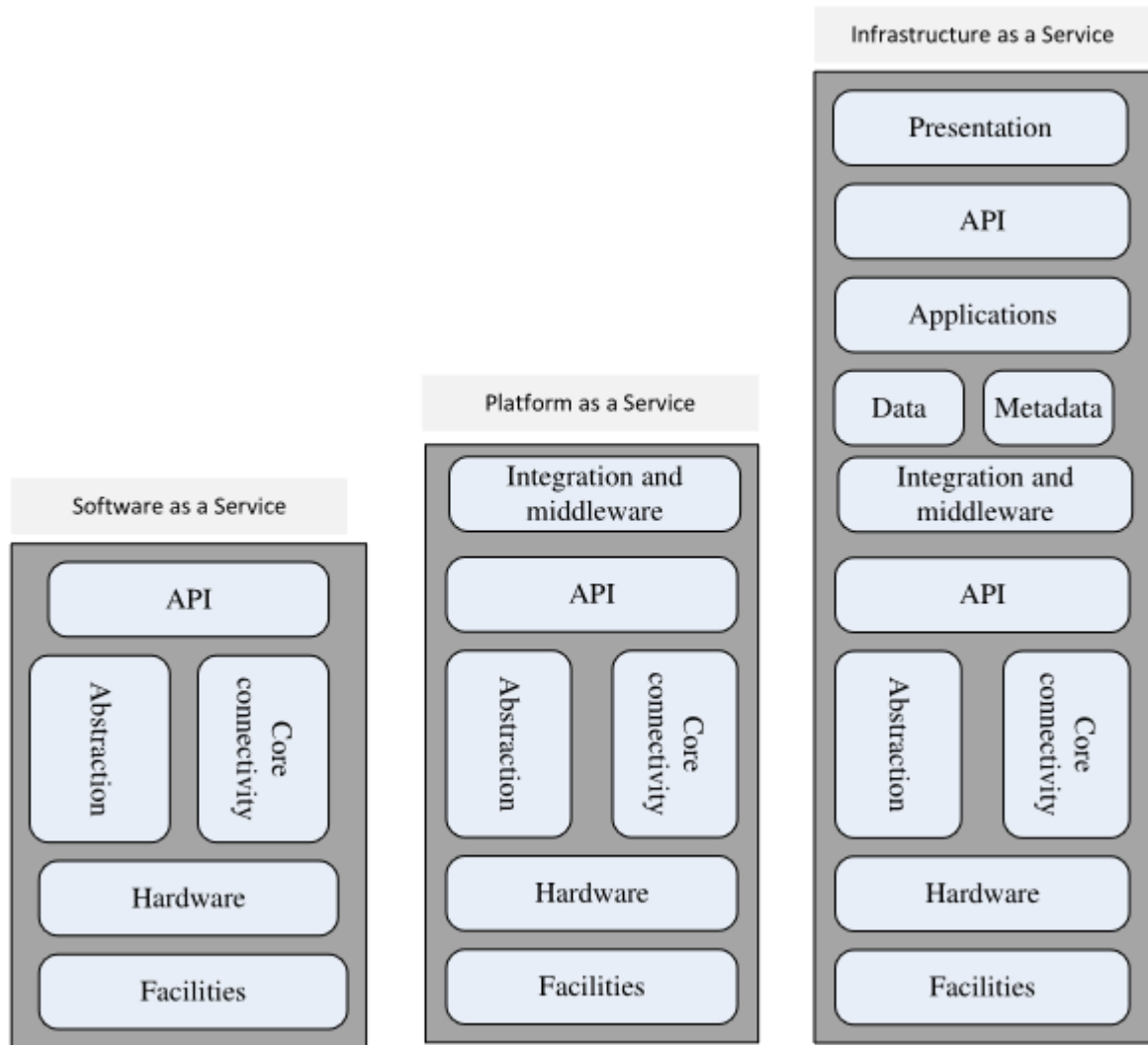
Παραδείγματα: Google Workspace (Gmail, Drive), Microsoft 365, Dropbox.

Οι υπηρεσίες σύννεφου (cloud services)



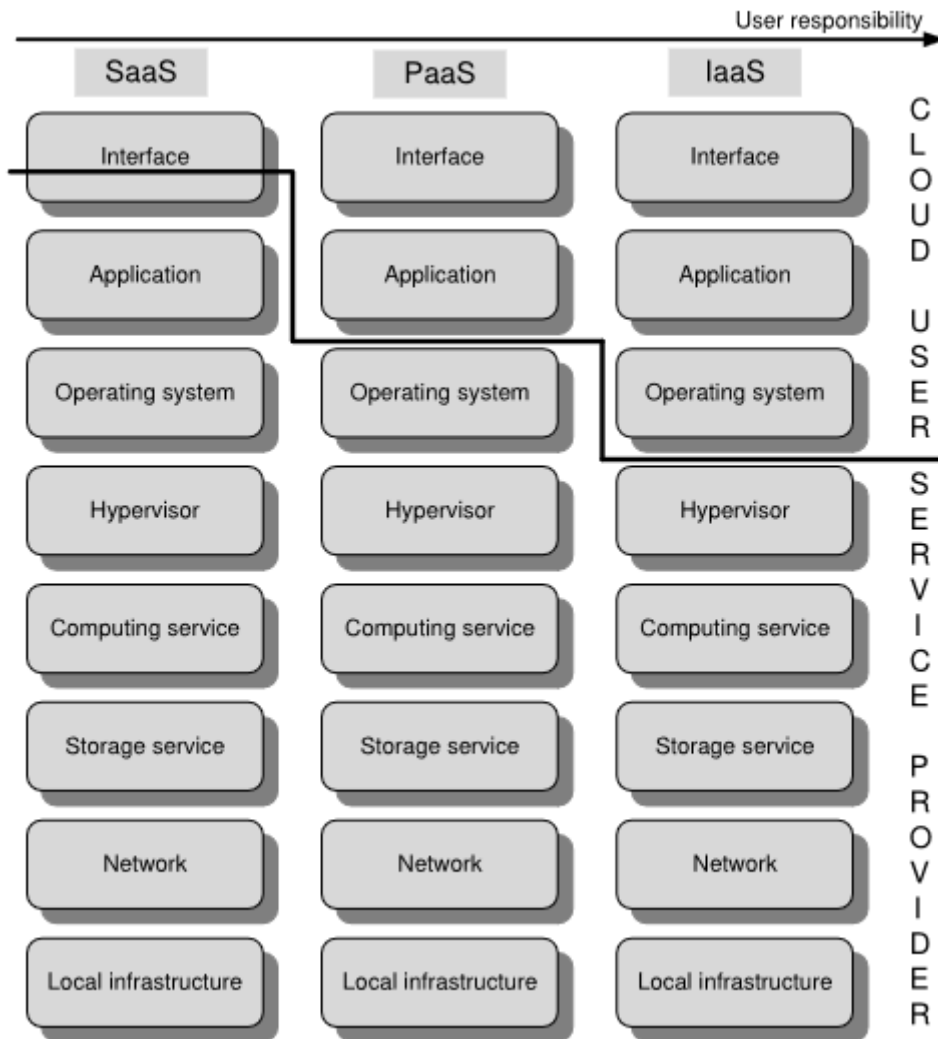
Σχήμα 16. Παραδείγματα ανά κατηγορία υπηρεσιών cloud [12]

Οι μεγαλύτεροι πάροχοι cloud υπηρεσιών είναι η Amazon με το Amazon Web Services (AWS), η Microsoft με το Microsoft Azure, η Google με το Google Cloud Platform (GCP) η IBM και η Oracle.



Σχήμα 17. Η δομή των τριών μοντέλων υπηρεσιών cloud, SaaS, PaaS και IaaS [10].

Το SaaS δίνει στους χρήστες τη δυνατότητα χρήσης εφαρμογών που παρέχονται από τον πάροχο υπηρεσιών, αλλά δεν επιτρέπει τον έλεγχο της πλατφόρμας ή της υποδομής. PaaS δίνει τη δυνατότητα ανάπτυξης εφαρμογών που έχουν δημιουργηθεί ή αποκτηθεί από τους καταναλωτές χρησιμοποιώντας γλώσσες προγραμματισμού και εργαλεία που υποστηρίζονται από τον πάροχο. Το IaaS επιτρέπει στο χρήστη να αναπτύξει και να εκτελέσει αυθαίρετο λογισμικό, το οποίο μπορεί να περιλαμβάνει λειτουργικά συστήματα και εφαρμογές.



Σχήμα 18. Τα όρια ευθύνης ανάμεσα στον χρήστη cloud υπηρεσιών και τον πάροχο cloud υπηρεσιών [10]

4.4. Μοντέλα με βάση τους αποδέκτες της υπηρεσίας

[10] Οι υπηρεσίες cloud μπορούν να κατηγοριοποιηθούν με βάση τις ομάδες χρήσης των υπηρεσιών ως εξής:

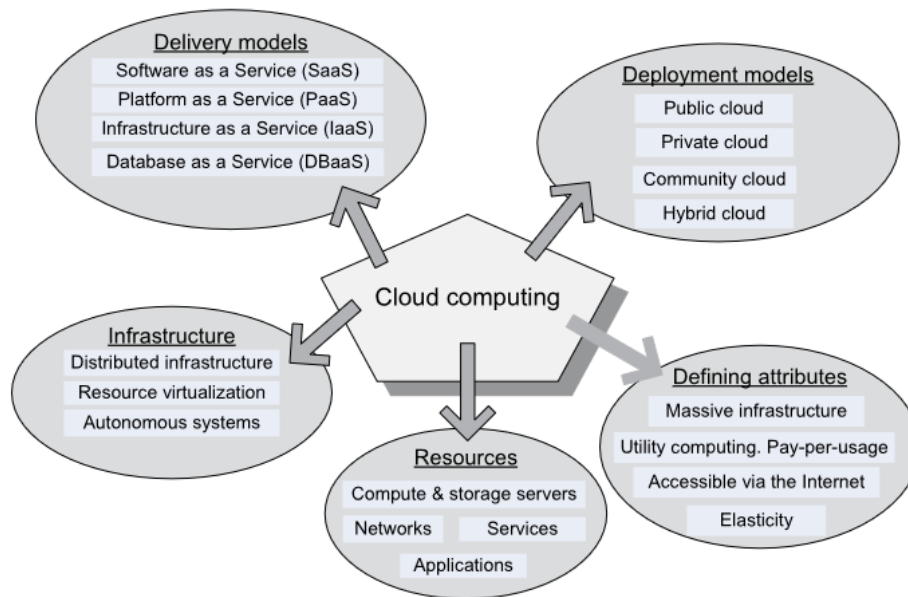
Private cloud. Η υποδομή cloud παρέχεται για αποκλειστική χρήση από έναν μόνο οργανισμό που περιλαμβάνει πολλούς καταναλωτές (π.χ. επιχειρηματικές μονάδες). Η ιδιοκτησία και η διαχείριση της υποδομής μπορεί να γίνεται από την ίδια την εταιρεία είναι υπάρχει ένας τρίτος οργανισμός στον οποίο ανήκει η υποδομή ή να της έχει αποδοθεί ο ρόλος της διαχείρισης ή μπορεί να υπάρχει συνδυασμός και των δύο. Το δίκτυο αυτό μπορεί να βρίσκεται εντός μια χώρας ή να εκτείνεται σε περισσότερες της μιας χώρας.

Community cloud. Η υποδομή cloud παρέχεται για αποκλειστική χρήση από συγκεκριμένη κοινότητα καταναλωτών από οργανισμούς που έχουν κοινές ανησυχίες (π.χ. αποστολή, απαιτήσεις ασφαλείας, πολιτικές και ζητήματα συμμόρφωσης). Μπορεί να είναι ιδιοκτησία και να διοικείται από έναν ή περισσότερους από τους οργανισμούς στην κοινότητα, ένα τρίτο

ενδιάμεσο ή να υπάρχει κάποιος συνδυασμός αυτών και μπορεί να υπάρχει εντός ή εκτός εγκαταστάσεων του οργανισμού.

Public cloud. Η υποδομή cloud παρέχεται για ανοιχτή χρήση από το ευρύ κοινό. Μπορεί να ανήκει και να γίνεται η διαχείριση από επιχείρηση, ακαδημαϊκό ή κυβερνητικό οργανισμό, ή κάποιο συνδυασμό τους. Υπάρχει στις εγκαταστάσεις του παρόχου cloud.

Hybrid cloud. Η υποδομή cloud είναι μια σύνθεση δύο ή περισσότερων διακριτών cloud υποδομών (ιδιωτικές, κοινοτικές ή δημόσιες) που παραμένουν μοναδικές οντότητες, αλλά είναι δεσμευμένες μαζί με τυποποιημένη ή αποκλειστική τεχνολογία που επιτρέπει την φορητότητα των δεδομένων και των εφαρμογών.



Σχήμα 19. Cloud computing: μοντέλα παράδοσης, μοντέλα ανάπτυξης, καθορισμός χαρακτηριστικών, πόρων και οργάνωση της υποδομής [10]

Κεφάλαιο 5

Διαδίκτυο των πραγμάτων (Internet of Things) και σύγχρονες τεχνολογικές λύσεις που εφαρμόζονται στο χώρο της Υγείας

5.1. Ορισμός του Διαδικτύου των πραγμάτων (Internet of Things) Το Διαδίκτυο των Πραγμάτων (IoT) είναι ένα αναδυόμενο θέμα τεχνικής, κοινωνικής και οικονομικής σπουδαιότητας. Καταναλωτικά προϊόντα, διαρκή αγαθά, αυτοκίνητα και φορτηγά, βιομηχανικά εξαρτήματα και εγκαταστάσεις, αισθητήρες και άλλα καθημερινά αντικείμενα συνδυάζονται με συνδεσιμότητα στο διαδίκτυο και με ισχυρές δυνατότητες ανάλυσης δεδομένων που υπόσχονται να αλλάξουμε τον τρόπο που ζούμε και εργαζόμαστε. [13]

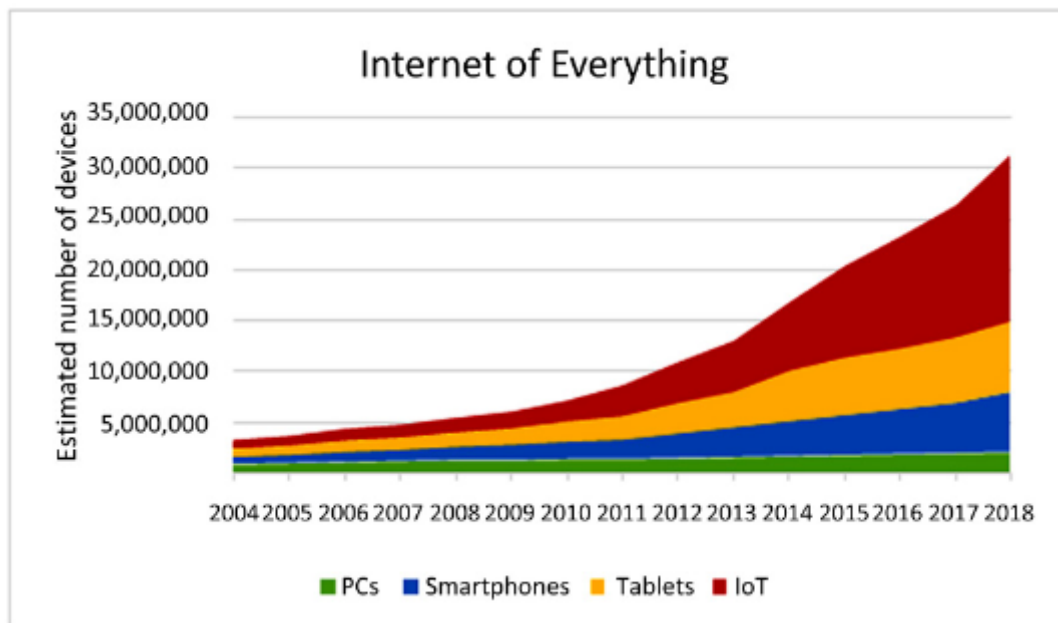
Μια σημαντική αλλαγή στις καθημερινές μας συνήθειες μπορεί να παρατηρηθεί μαζί με την ευρεία εφαρμογή συσκευών και τεχνολογιών IoT. Το IoT είναι παντού, αν και δεν το βλέπουμε πάντα ή δεν γνωρίζουμε ότι μια συσκευή είναι μέρος του. Για τους καταναλωτές, νέα προϊόντα IoT όπως συσκευές με δυνατότητα σύνδεσης στο διαδίκτυο, εξαρτήματα οικιακού αυτοματισμού και συσκευές διαχείρισης ενέργειας μας οδηγούν στο να δούμε το «Έξυπνο σπίτι», το οποίο παρέχει περισσότερη ασφάλεια και ενεργειακή απόδοση. Άλλες προσωπικές συσκευές IoT, όπως τα ρολόγια φυσικής κατάστασης και υγείας που υποστηρίζουν τις ιατρικές συσκευές με διασύνδεση δικτύου, αλλάζουν τον τρόπο παροχής των υπηρεσιών υγειονομικής περίθαλψης. Το Διαδίκτυο των Πραγμάτων μετατρέπει τα φυσικά αντικείμενα σε ένα οικοσύστημα πληροφοριών που μοιράζεται μεταξύ φορητών ή ακόμη και εμφυτευσιμων συσκευών, καθιστώντας την τεχνολογία και τα δεδομένα της ζωής μας πλούσια.

Η τεχνολογία IoT υπόσχεται να είναι χρήσιμη για άτομα με αναπηρίες και ηλικιωμένους, επιτρέποντας βελτιωμένα επίπεδα ανεξαρτησίας και ποιότητας ζωής με λογικό κόστος. Συστήματα Διαδικτύου των Πραγμάτων όπως δικτυωμένα οχήματα, έξυπνα συστήματα κυκλοφορίας και αισθητήρες ενσωματωμένοι σε δρόμους και γέφυρες μας φέρνουν πιο κοντά στην ιδέα των «έξυπνων πόλεων», που συμβάλλουν στη μείωση της συμφόρησης και της κατανάλωσης ενέργειας. Η τεχνολογία IoT προσφέρει τη δυνατότητα να μεταμορφώσει τη γεωργία, τη βιομηχανία και τη παραγωγή και διανομή ενέργειας αυξάνει τη διαθεσιμότητα πληροφοριών κατά μήκος της αλυσίδας αξίας παραγωγής χρησιμοποιώντας δικτυωμένους αισθητήρες.

Ορισμένες εταιρείες και ερευνητικοί Οργανισμοί έχουν παράσχει ένα ευρύ φάσμα προσδοκιών σχετικά με τον πιθανό αντίκτυπο του Διαδικτύου των Πραγμάτων στο διαδίκτυο και στην οικονομία την επόμενη δεκαετία. Η Huawei αναμένει 100 δισεκατομμύρια συνδέσεις IoT έως το 2025. Η εκτίμηση του δυνητικού οικονομικού αντίκτυπου του Διαδικτύου των Πραγμάτων είναι από 3,9 έως 11 τρισεκατομμύρια δολάρια ετησίως το 2025, λόγω του χαμηλότερου κόστους των συσκευών, του προηγμένου υπολογιστικού χώρου αποθήκευσης cloud, της υψηλότερης ταχύτητας και του χαμηλότερου κόστους παράδοσης. Αυτό αυξάνει τον αριθμό των μηχανών και συσκευών συνδεδεμένο στο διαδίκτυο. Εκτιμάται επίσης ότι το Διαδίκτυο των Πραγμάτων θα συνεισφέρει από 4% έως 11% του παγκόσμιου ΑΕΠ το 2025.

Το "Πράγμα" στο Διαδίκτυο των Πραγμάτων μπορεί να είναι οποιαδήποτε συσκευή με οποιονδήποτε τύπο αισθητήρα ενσωματωμένο με δυνατότητα συλλογής δεδομένων και μετάδοσης τους μέσω του δικτύου χωρίς χειροκίνητη παρέμβαση. Η τεχνολογία που είναι ενσωματωμένη στο αντικείμενο βοηθά στην αλληλεπίδραση με τις εσωτερικές καταστάσεις και το εξωτερικό περιβάλλον, το οποίο με τη σειρά του βοηθά στη διαδικασία λήψης αποφάσεων.

Υπάρχει επίσης μεγάλη γκάμα τρόπων διασύνδεσης των συσκευών αυτών στο διαδίκτυο όπως το Wi-Fi, NB-IoT, Bluetooth Low Energy (BLE), 5G, LoRaWAN. Οι συσκευές αυτές διαθέτουν αισθητήρες για να καταγράφουν δεδομένα θερμοκρασίας, κίνησης, τοποθεσίας κλπ. Συνήθως βρίσκονται σε εξωτερικά περιβάλλοντα στα οποία δεν υπάρχουν πηγές ηλεκτρικής ενέργειας και έτσι είναι σχεδιασμένες για χαμηλή κατανάλωση ενέργειας και έχουν μεγάλη αυτονομία.



Σχήμα 20. Πρόβλεψη εξάπλωσης IoT συσκευών

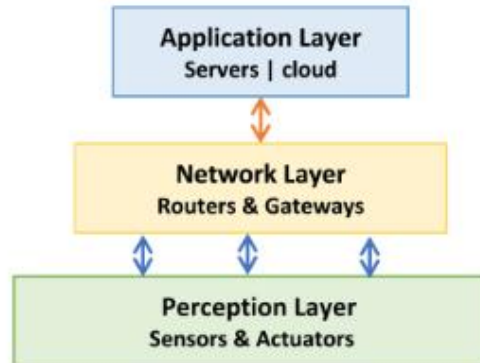
5.2 Η Αρχιτεκτονική του IoT

Το κύριο πρόβλημα με το Διαδίκτυο των Πραγμάτων είναι ότι είναι πολύ ευρύ και απεριόριστο, επομένως η υλοποίηση της ιδέας του εξαρτάται ουσιαστικά από την αρχιτεκτονική του. Στο αρχικό στάδιο της έρευνας, εισήχθη η αρχιτεκτονική των τριών επιπέδων.

5.2.1 Η Αρχιτεκτονική των τριών επιπέδων

Αυτή η αρχιτεκτονική αποτελείται από τρία επίπεδα. Το πρώτο είναι το επίπεδο αντίληψης που είναι το φυσικό στρώμα. Διαθέτει αισθητήρες για την ανίχνευση και τη συλλογή πληροφοριών για το περιβάλλον. Αισθάνεται κάποιες φυσικές παραμέτρους ή προσδιορίζει άλλα έξυπνα αντικείμενα στο περιβάλλον. Το δεύτερο είναι το επίπεδο δικτύου που είναι υπεύθυνο για τη σύνδεση με άλλα έξυπνα πράγματα, συσκευές δικτύου και διακομιστές. Χαρακτηρίζεται

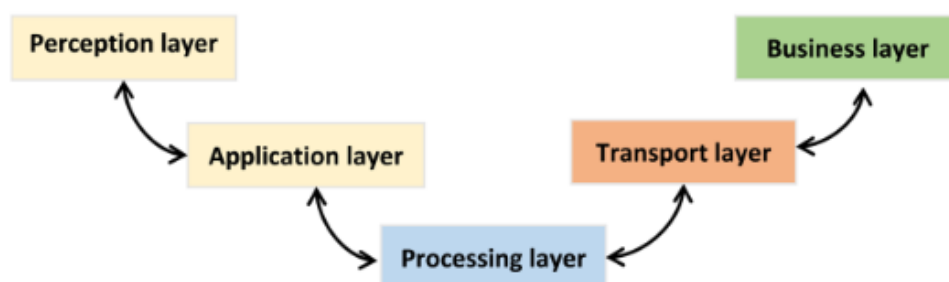
επίσης από τη μετάδοση και την επεξεργασία των δεδομένων αισθητήρων. Το τρίτο είναι το επίπεδο εφαρμογής που είναι υπεύθυνο για την παροχή συγκεκριμένων υπηρεσιών εφαρμογής στον χρήστη. Ορίζει διάφορες εφαρμογές στις οποίες μπορεί να αναπτυχθεί το Διαδίκτυο των Πραγμάτων όπως έξυπνα σπίτια, έξυπνες πόλεις, έξυπνη υγεία, έξυπνη γεωργία κ.λ.π.



Σχήμα 21. Η Αρχιτεκτονική των τριών επιπέδων

5.2.2 Η Αρχιτεκτονική των πέντε επιπέδων

Η αρχιτεκτονική πέντε επιπέδων έχει σχεδιαστεί για να καθορίσει την πλήρη έννοια της λειτουργίας και της ανάπτυξης συσκευών IoT. Η νέα δομή περιλαμβάνει 5 στρώσεις. Τα επίπεδα αντίληψης και εφαρμογής λειτουργούν με τον ίδιο τρόπο όπως στην προηγούμενη αρχιτεκτονική. Το κύριο καθήκον του επιπέδου επεξεργασίας είναι να επεξεργάζεται τις πληροφορίες που λαμβάνονται από το επίπεδο δικτύου και να λαμβάνει αποφάσεις με βάση τα αποτελέσματα που επιτυγχάνονται από την πανταχού παρούσα πληροφορική. Το επίπεδο μεταφοράς που μεταφέρει δεδομένα αισθητήρα από το επίπεδο αντίληψης στο επίπεδο επεξεργασίας και αντίστροφα μέσω δικτύων όπως ασύρματα, LAN, 5G, LTE, RFID και Bluetooth. Τέλος, το επιχειρησιακό επίπεδο οπτικοποιεί πληροφορίες και στατιστικά στοιχεία από το επίπεδο εφαρμογής και χρησιμοποιεί αυτή τη γνώση για να σχεδιάσει μελλοντικούς στόχους και στρατηγικές.



Σχήμα 22. Η Αρχιτεκτονική των πέντε επιπέδων

5.3 Αισθητήρες & ενεργοποιητές

Οι αισθητήρες και οι ενεργοποιητές είναι μεταξύ των δομικών στοιχείων του IoT. Σε πολλές εφαρμογές IoT, χρειάζεστε έναν ή περισσότερους αισθητήρες για τη συλλογή δεδομένων και πληροφοριών σχετικά με το σύστημα. Τα δεδομένα υποβάλλονται σε επεξεργασία και ενδέχεται να εκδοθούν εντολές στους ενεργοποιητές που με τη σειρά τους επηρεάζουν το

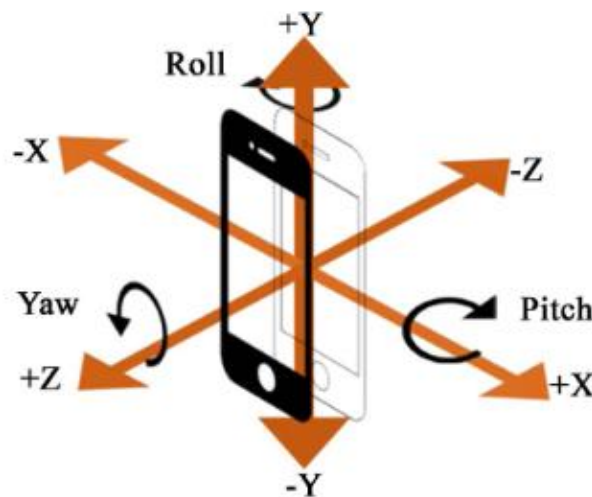
σύστημα και με άλλο τρόπο οι αισθητήρες συλλέγουν τα δεδομένα που θα μεταδοθούν μέσω του δικτύου και τους ενεργοποιητές που επιτρέπουν να λειτουργούν πράγματα όπως: Οι αισθητήρες υγρασίας παρέχουν δεδομένα για τον έλεγχο των συστημάτων άρδευσης. Οι αισθητήρες κυκλοφορίας παρέχουν δεδομένα για τον έλεγχο των φωτεινών σηματοδοτών. Οι αισθητήρες πληρότητας παρέχουν δεδομένα για τον έλεγχο των κτιρίων. Οι αισθητήρες και οι ενεργοποιητές επιτρέπουν λύσεις IoT σε κάθε κατακόρυφο πεδίο IoT, από έξυπνες πόλεις έως έξυπνη γεωργία και από την προσωπική υγεία έως τις έξυπνες μεταφορές.

5.3.1 Αισθητήρες κινητών τηλεφώνων

[13] Στη σύγχρονη ζωή, υπάρχουν πολλά σημαντικά στοιχεία χωρίς τα οποία δεν μπορούμε να ζήσουμε και βρισκόμαστε σε μεγαλύτερη ανάγκη, καθώς κάνουν τη ζωή μας πιο εύκολη και ομαλή. Το smartphone είναι ο πόλος και το στοιχείο της ζωής μας.

Υπάρχουν πολλά που μπορούμε να κάνουμε με ένα smartphone και πολλούς διαφορετικούς τρόπους οι οποίοι παίζουν σοβαρό ρόλο στη ζωή του καθενός. Πολλά από τα πιο εντυπωσιακά επιτεύγματα επιτυγχάνονται μέσω διαφόρων αισθητήρων που βρίσκονται στα κινητά τηλέφωνα. Εδώ είναι τα πιο σημαντικά:

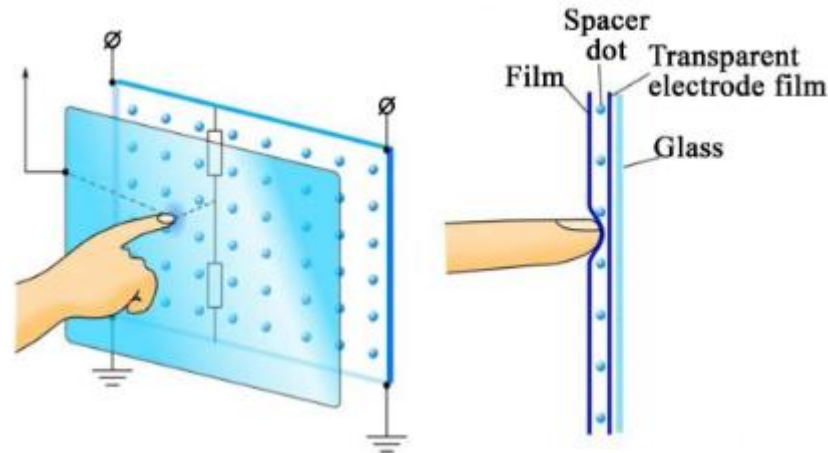
1. **Επιταχυνσιόμετρο:** Ο αισθητήρας επιταχυνσιόμετρο ανιχνεύει την επιτάχυνση, τους κραδασμούς και την κλίση για να προσδιορίσει την κίνηση και τον ακριβή προσανατολισμό στις τρεις διαστάσεις. Οι εφαρμογές χρησιμοποιούν αυτόν τον αισθητήρα έξυπνου τηλεφώνου για να προσδιορίσουν εάν το τηλέφωνό σας είναι σε κατακόρυφο ή οριζόντιο προσανατολισμό. Μπορεί επίσης να πει αν η οθόνη του τηλεφώνου σας είναι στραμμένη προς τα πάνω ή προς τα κάτω. Τα μοτίβα δεδομένων που καταγράφονται από το επιταχυνσιόμετρο μπορούν να χρησιμοποιηθούν για την ανίχνευση φυσικών δραστηριοτήτων του χρήστη όπως τρέξιμο, περπάτημα και ποδήλατο.



Σχήμα 23. Επιταχυνσιόμετρο

2. **Γυροσκόπιο:** Το γυροσκόπιο παρέχει λεπτομέρειες προσανατολισμού και κατεύθυνση, όπως πάνω-κάτω και αριστερά-δεξιά, αλλά με μεγαλύτερη ακρίβεια, όπως η κλίση της συσκευής. Άρα το γυροσκόπιο έχει την ικανότητα να μετράει την περιστροφή. Ως εκ τούτου, μπορεί να πει πόσο έχει περιστραφεί ένα έξυπνο τηλέφωνο και προς ποια

- κατεύθυνση. Η εφαρμογή Google Sky Map χρησιμοποιεί αισθητήρα γυροσκόπιου για να προσδιορίσει την κατεύθυνση προς την οποία είναι στραμμένο το τηλέφωνό σας.
3. Μαγνητόμετρο: Συνήθως γνωστό ως πυξίδα, μπορεί να ανιχνεύσει μαγνητικά πεδία, επομένως η εφαρμογή Compass στα smartphone χρησιμοποιεί αυτόν τον αισθητήρα για να δείχνει προς τον βόρειο πόλο του πλανήτη. Αυτός ο έξυπνος αισθητήρας χρησιμοποιείται σε ανιχνευτή μετάλλων και μπορείτε να τον βρείτε κάθε φορά που ανοίγετε τους Χάρτες Google ή γενικότερα οποιαδήποτε εφαρμογή Χαρτών. Το μαγνητόμετρο στεγάζεται σε ένα μικρό ηλεκτρονικό τσιπ που συχνά περιλαμβάνει έναν άλλο αισθητήρα και είναι συνήθως ενσωματωμένο στο επιταχυνσιόμετρο όπου βοηθά στη διόρθωση των αρχικών μαγνητικών μετρήσεων χρησιμοποιώντας πληροφορίες κλίσης από τον βοηθητικό αισθητήρα.
 4. GPS: Στο Global Positioning System GPS δηλαδή στο Παγκόσμιο Σύστημα Εντοπισμού, τα κινητά τηλέφωνα επικοινωνούν με τους δορυφόρους για να προσδιορίσουν με ακρίβεια τη θέση μας στη Γη. Η τεχνολογία GPS δεν χρησιμοποιεί στην πραγματικότητα δεδομένα Διαδικτύου, γι' αυτό μόλις ανοίξουμε την Εφαρμογή μπορούμε να βρούμε την τοποθεσία μας στους χάρτες ακόμη και εκτός σύνδεσης του Δικτύου, αλλά ο ίδιος ο χάρτης είναι θολός καθώς απαιτεί το δίκτυο για τη φόρτωση λεπτομερειών. Το GPS χρησιμοποιείται σε όλες τις εφαρμογές που βασίζονται στην τοποθεσία. Επιταχυνσιόμετρο, γυροσκόπιο, μαγνητόμετρο και GPS συνεργάζονται για να δημιουργήσετε την τέλεια πλοήγηση στο smartphone σας.
 5. Μικρόφωνο: Το μικρόφωνο είναι βασικά ένας αισθητήρας ήχου που ανιχνεύει και μετρά την ένταση του ήχου. Τα smartphones γενικά χρησιμοποιούν ηλεκτρικά μικρόφωνα μικρού μεγέθους
 6. Αισθητήρας φωτός περιβάλλοντος: Ο αισθητήρας φωτός ανιχνεύει τα επίπεδα φωτισμού στην περιοχή για να προσαρμόσει ανάλογα τη φωτεινότητα της οθόνης. Χρησιμοποιείται στην αυτόματη ρύθμιση φωτεινότητας για μείωση ή αύξηση της φωτεινότητας της οθόνης smartphone με βάση τη διαθεσιμότητα φωτός. Η μείωση της φωτεινότητας της οθόνης σε μια κινητή συσκευή παρατείνει επίσης τη διάρκεια ζωής της μπαταρίας.
 7. Αισθητήρες οθόνης αφής: Οι αισθητήρες του έξυπνου τηλεφώνου στην οθόνη αφής περιέχουν ηλεκτρικό ρεύμα που τη διαπερνά συνεχώς και το άγγιγμα της οθόνης προκαλεί αλλαγή στα σήματα. Αυτή η αλλαγή αποτελεί είσοδο στη συσκευή. Σήμερα, όλα τα smartphone χρησιμοποιούν αυτήν την τεχνολογία οθόνης. Βασικά, η οθόνη αφής είναι υπεύθυνη για τις βασικές λειτουργίες εισόδου και εξόδου και χρησιμοποιείται για το πάτημα και τη σύνταξη γραμμάτων. Η οθόνη αφής περιέχει τρεις κύριες ενέργειες αλληλεπίδρασης:
 - Η κύρια δραστηριότητα και στόχος της οθόνης αφής είναι το άγγιγμα ή το πάτημα ορίζεται ως η διαδικασία του κλικ στην οθόνη σε οποιοδήποτε σημείο για να ανοίξει, να κλείσει ή να πληκτρολογήσει έναν χαρακτήρα.
 - Το Multi-touch ορίζεται ως η διαδικασία του αγγίγματος της οθόνης με περισσότερα από ένα δάχτυλα ταυτόχρονα και αυτή η λειτουργία χρησιμοποιείται συνήθως σε εφαρμογές gaming.
 - Η χειρονομία ορίζεται ως η διαδικασία σχεδίασης ενός συγκεκριμένου μοτίβου στην οθόνη αφής. Οι χειρονομίες μπορούν να εφαρμοστούν με ένα δάχτυλο ως μεταφορά και απόθεση ή με πολλά δάχτυλα, όπως κατά τη διαδικασία επεξεργασίας των φωτογραφιών αλλαγής μεγέθους και αλλαγής του ζουμ της κάμερας.



Σχήμα 24. Αισθητήρας Οθόνης Αφής

8. Αισθητήρας δακτυλικών αποτυπωμάτων: Ο αισθητήρας δακτυλικών αποτυπωμάτων επιτρέπει τη βιομετρική επαλήθευση για την ασφάλεια πολλών smartphone σήμερα. Είναι ένας χωρητικός σαρωτής που καταγράφει ηλεκτρικά το δακτυλικό αποτύπωμα του χρήστη. Όταν τοποθετείτε το δάχτυλό σας στην επιφάνειά του, οι άκρες των δακτυλικών σας αποτυπωμάτων αγγίζουν την επιφάνεια ενώ υπάρχει ένα μικρό κενό μεταξύ των υποδοχών μεταξύ των άκρων. Με λίγα λόγια, μετρά τις αποστάσεις και το ανομοιόμορφο σχέδιο μεταξύ των άκρων στην επιφάνεια του δακτύλου σας. Αυτός ο αισθητήρας smartphone είναι πολύ χρήσιμος σε εφαρμογές που απαιτούν έλεγχο ταυτότητας, όπως εφαρμογές πληρωμής μέσω κινητού
9. Αισθητήρας καρδιακών παλμών: Ο αισθητήρας καρδιακών παλμών μετρά τον καρδιακό παλμό με τη βοήθεια οπτικών αισθητήρων και λυχνιών LED. Ένα φως LED εκπέμπεται προς το δέρμα και αυτός ο αισθητήρας smartphone ανιχνεύει τα κύματα φωτός που αντανακλώνται σε αυτό. Υπάρχει διαφορά στην ένταση του φωτός όταν υπάρχει παλμός. Ο καρδιακός ρυθμός μετριέται με τον υπολογισμό των αλλαγών στην ένταση του φωτός μεταξύ των λεπτών παλμών των αιμοφόρων αγγείων. Πολλές εφαρμογές γυμναστικής και υγείας χρησιμοποιούν αυτήν τη μέθοδο για να υπολογίσουν τον καρδιακό σας ρυθμό.
10. Αισθητήρας Barcode/QR Code: Τα περισσότερα smartphone διαθέτουν αισθητήρες γραμμικού κώδικα που μπορούν να διαβάσουν έναν γραμμωτό κώδικα ανιχνεύοντας το φως που ανακλάται από τον κώδικα. Παράγει ένα αναλογικό σήμα με μεταβλητή τάση που αντιπροσωπεύει τον γραμμωτό κώδικα. Στη συνέχεια αυτό το αναλογικό σήμα μετατρέπεται σε ψηφιακό σήμα και τελικά αποκωδικοποιείται για να αποκαλύψει τις πληροφορίες σε αυτό. Οι αισθητήρες γραμμωτού κώδικα είναι χρήσιμοι για τη σάρωση προϊόντων γραμμωτού κώδικα ή κωδικών QR. Χρησιμοποιείται στις περισσότερες εφαρμογές μέσω κοινωνικής δικτύωσης και είναι πολύ χρήσιμο στη διαδικασία πληρωμής.

5.3.2 Ιατρικοί αισθητήρες

Το Διαδίκτυο των Πραγμάτων διαδραματίζει σημαντικότερο ρόλο στην ιατρική τεχνολογία με στόχο να γίνουν οι ιατροτεχνολογικές συσκευές πιο αποτελεσματικές και ασφαλέστερες, απλοποιώντας παράλληλα τη λειτουργία τους. Το Internet of Things επεκτείνει τον ιατρικό τομέα με πολλές εφαρμογές που βασίζονται σε έξυπνους αισθητήρες, οι οποίοι παρακολουθούν την υγεία ενός ασθενούς όταν δεν βρίσκεται στο νοσοκομείο ή όταν είναι μόνος. Μετά από αυτό, μπορούν να παρέχουν άμεση ανατροφοδότηση στον γιατρό, τους συγγενείς ή τον ασθενή. Στην ιατρική αγορά, μπορείτε να βρείτε πολλούς φορητούς αισθητήρες. Μπορούν να διαγνώσουν τον ασθενή και να μετρήσουν διάφορες παραμέτρους όπως καρδιακό ρυθμό, αναπνευστικό ρυθμό, πίεση αίματος, επίπεδα σακχάρου στο αίμα και θερμοκρασία σώματος. Το Internet of Things προσπαθεί να αποκτήσει τη νέα του συσκευή που ξεπέρασε τις έξυπνες συσκευές που φοράει ο χρήστης, αυτή τη συσκευή για να ανιχνεύει σημεία που είναι επικολλημένα στο δέρμα. Τέτοια τατουάζ στο δέρμα είναι μιας χρήσης, ελαστικά και πολύ φθηνά. Η ηλεκτρονική εφαρμογή είναι περίβλημα από καουτσούκ. Ο ασθενής υποτίθεται ότι φοράει αυτά τα επιθέματα για μερικές ημέρες για να παρακολουθεί συνεχώς ένα ζωτικό εργαστήριο υγείας.



Σχήμα 25. Έξυπνα ρολόγια και ιχνηλάτες γυμναστικής στα αριστερά. Ενσωματωμένο έμπλαστρο δέρματος στα δεξιά

5.4. Κύρια ζητήματα και προκλήσεις του IoT

Η κοινή χρήση συστημάτων που βασίζονται στο IoT σε όλες τις πτυχές της ανθρώπινης ζωής και οι διάφορες τεχνολογίες που εμπλέκονται στη μεταφορά δεδομένων μεταξύ ενσωματωμένων συσκευών το έκαναν περίπλοκο και οδήγησαν σε πολλά προβλήματα και προκλήσεις. Αυτά περιλαμβάνουν την ασφάλεια, την μυστικότητα, την διαλειτουργικότητα, την δημιουργία προτύπων ενώ παράλληλα αναδύονται νομικά δέματα και θέματα προσωπικών δεδομένων.

5.4.1. Ασφάλεια

Αν και τα ζητήματα ασφαλείας δεν είναι νέα στο πλαίσιο της πληροφορικής, τα χαρακτηριστικά πολλών εφαρμογών IoT παρουσιάζουν νέες και μοναδικές προκλήσεις ασφαλείας. Η αντιμετώπιση αυτών των προκλήσεων και η διασφάλιση της ασφαλείας στα προϊόντα και τις υπηρεσίες IoT θα πρέπει να είναι πρωταρχική προτεραιότητα και οι χρήστες

πρέπει να αισθάνονται ότι οι συσκευές IoT και οι σχετικές υπηρεσίες δεδομένων προστατεύονται από τρωτά σημεία, ειδικά καθώς αυτή η τεχνολογία έχει γίνει πιο διάχυτη και ενσωματωμένη στην καθημερινότητά μας. Οι συσκευές και οι υπηρεσίες IoT με κακή ασφάλεια μπορούν να λειτουργήσουν ως πιθανά σημεία εισόδου για επίθεση στον κυβερνοχώρο και να εκθέσουν τα δεδομένα των χρηστών σε κλοπή αφήνοντας τη ροή δεδομένων ανεπαρκώς προστατευμένη. Η διασυνδεδεμένη φύση των συσκευών IoT σημαίνει ότι κάθε μη ασφαλής συσκευή που είναι συνδεδεμένη στο Διαδίκτυο έχει τη δυνατότητα να επηρεάσει την ασφάλεια και την ανθεκτικότητα του Διαδικτύου παγκοσμίως. Αυτή η πρόκληση ενισχύεται από άλλους παράγοντες, όπως η ευρεία ανάπτυξη ομοιογενών συσκευών IoT, η ικανότητα ορισμένων συσκευών να συνδέονται αυτόματα με άλλες και η δυνατότητα ανάπτυξης αυτών των συσκευών σε μη ασφαλή περιβάλλοντα.

5.4.2. Ιδιωτικότητα

Το πλήρες δυναμικό του Διαδικτύου των πραγμάτων εξαρτάται από στρατηγικές που σέβονται τις μεμονωμένες επιλογές απορρήτου σε ένα ευρύ φάσμα προσδοκιών. Οι ροές δεδομένων και το απόρρητο των χρηστών που παρέχουν οι συσκευές IoT μπορούν να δημιουργήσουν απίστευτη και μοναδική αξία για τους χρήστες του IoT, αλλά οι ανησυχίες σχετικά με το απόρρητο και τις πιθανές βλάβες μπορεί να εμποδίσουν την πλήρη υιοθέτηση του IoT. Αυτό σημαίνει ότι τα δικαιώματα απορρήτου και η απαίτηση για σεβασμό του απορρήτου των χρηστών είναι αναπόσπαστα στοιχεία για τη διασφάλιση της εμπιστοσύνης των χρηστών στο Διαδίκτυο, τις συνδεδεμένες συσκευές και τις σχετικές υπηρεσίες.

5.4.3. Διαλειτουργικότητα και Πρότυπα

Η διαλειτουργικότητα είναι η ικανότητα ανταλλαγής πληροφοριών μεταξύ διαφόρων συσκευών και συστημάτων IoT. Αυτή η ανταλλαγή πληροφοριών δεν βασίζεται σε δημοσιευμένο λογισμικό και υλικό. Το πρόβλημα της διαλειτουργικότητας προκύπτει λόγω της ετερογενούς φύσης της τεχνολογίας και των διαφόρων λύσεων που χρησιμοποιούνται για την ανάπτυξη του IoT.

Τα τέσσερα επίπεδα διαλειτουργικότητας είναι το τεχνικό, το σημασιολογικό, το συντακτικό και το οργανωτικό. Με τη διαλειτουργικότητα ως σημαντικό ζήτημα, οι ερευνητές έχουν συμφωνήσει με διάφορες λύσεις, όπως προσαρμοστική, βασισμένη σε πύλη, αρχιτεκτονική βασισμένη σε εικονικά δίκτυα και υπηρεσίες. Είναι επίσης γνωστές ως προσεγγίσεις για την αντιμετώπιση της διαλειτουργικότητας. Αν και οι μέθοδοι αντιμετώπισης της διαλειτουργικότητας ανακουφίζουν κάποια πίεση στα συστήματα IoT, εξακολουθούν να υπάρχουν ορισμένες προκλήσεις με τη δυνατότητα διαλειτουργικότητας που θα μπορούσαν να αποτελέσουν τομέα για μελλοντικές μελέτες.

5.4.4. Νομικά, Ρυθμιστικά και Δικαιώματα

Η χρήση συσκευών IoT εγείρει πολλά νέα ρυθμιστικά και νομικά ζητήματα εκτός από την ενίσχυση των υφιστάμενων νομικών ζητημάτων γύρω από το Διαδίκτυο. Τα ερωτήματα είναι ευρύτατα και ο γρήγορος ρυθμός αλλαγής στην τεχνολογία IoT συχνά ξεπερνά την προσαρμοστικότητα των σχετικών πολιτικών και των νομικών και ρυθμιστικών δομών. Με την ανάπτυξη του Διαδικτύου των Πραγμάτων, πολλά προβλήματα της πραγματικής ζωής έχουν λυθεί αλλά έχουν επίσης προκαλέσει κρίσιμες ηθικές και νομικές προκλήσεις όπως η ασφάλεια των δεδομένων, η προστασία της ιδιωτικής ζωής, η εμπιστοσύνη, η ασφάλεια και η

χρησιμότητα των δεδομένων. Έχει επίσης παρατηρηθεί ότι η πλειονότητα των χρηστών IoT απαιτεί κυβερνητικούς κανόνες και κανονισμούς σχετικά με την προστασία δεδομένων, το απόρρητο και την ασφάλεια λόγω δυσπιστίας προς τις συσκευές IoT. Ως εκ τούτου, αυτό το ζήτημα θα πρέπει να ληφθεί υπόψη για να διατηρηθεί και να βελτιωθεί η εμπιστοσύνη μεταξύ των ανθρώπων σχετικά με τη χρήση συσκευών και συστημάτων IoT.

5.5 Επικοινωνία

Το Internet of Things αποτελείται από πολλές έξυπνες συσκευές που επικοινωνούν μεταξύ τους. Αυτές οι συσκευές επιτρέπουν την ανταλλαγή και τη συλλογή δεδομένων. Οι έξυπνες συσκευές μπορούν να έχουν ενσύρματη ή ασύρματη σύνδεση. Συνήθως, οι συσκευές IoT συνδέονται στο Διαδίκτυο μέσω της στοίβας Πρωτοκόλλου Διαδικτύου (IP). Αυτός ο συνδυασμός είναι πολύ περίπλοκος και απαιτεί μεγάλη ποσότητα ενέργειας και μνήμης από τις συνδεδεμένες συσκευές. Αυτές οι συσκευές μπορούν επίσης να συνδεθούν τοπικά μέσω δικτύων NON-IP που καταναλώνουν λιγότερη ενέργεια και συνδέονται στο διαδίκτυο μέσω smart gateways.

5.5.1 Επικοινωνίες από συσκευή σε συσκευή

Ένα μοντέλο επικοινωνίας από συσκευή σε συσκευή αντιπροσωπεύει δύο ή περισσότερες συσκευές που επικοινωνούν απευθείας και επικοινωνούν μεταξύ τους, αντί για έναν ενδιάμεσο διακομιστή εφαρμογών. Αυτές οι συσκευές επικοινωνούν μέσω πολλών τύπων δικτύων, συμπεριλαμβανομένου του IP και του Διαδικτύου. Ωστόσο, αυτές οι συσκευές χρησιμοποιούν πρωτόκολλα όπως Bluetooth, Z-Wave ή ZigBee για τη δημιουργία απευθείας συνδέσεων από συσκευή σε συσκευή.

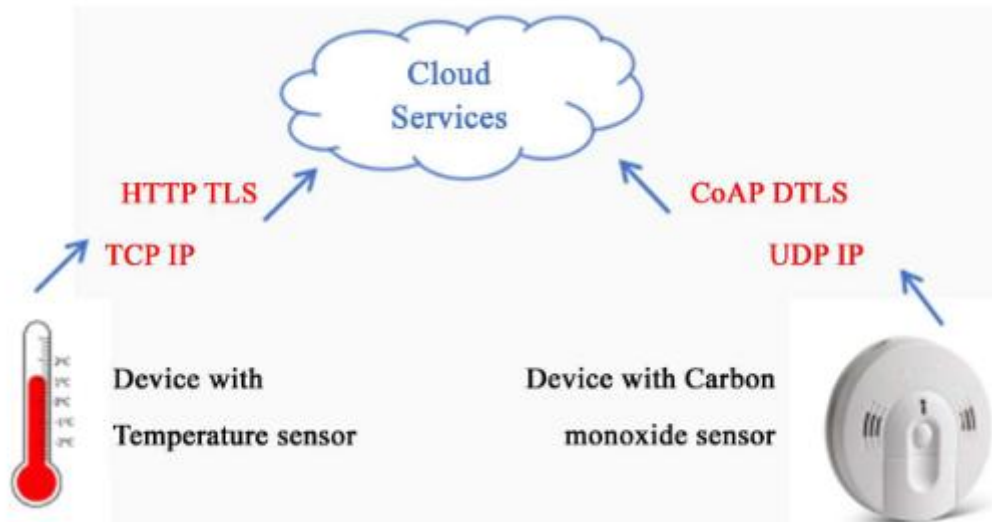
Τα δίκτυα συσκευής-συσκευής επιτρέπουν σε αυτές τις συσκευές που δεσμεύτηκαν σε μια συγκεκριμένη σύνδεση ένα πρωτόκολλο επικοινωνίας και ανταλλαγής μηνυμάτων για να επιτύχουν την αποστολή τους. Αυτή είναι η φόρμα επικοινωνίας που χρησιμοποιείται συνήθως σε εφαρμογές όπως τα συστήματα οικιακού αυτοματισμού. Συνήθως, τα μικρά πακέτα δεδομένων χρησιμοποιούνται πληροφορίες για την επικοινωνία μεταξύ συσκευών με σχετικά χαμηλές απαιτήσεις ρυθμού δεδομένων.



Σχήμα 26. Επικοινωνία από συσκευή σε συσκευή

5.5.2 Επικοινωνίες από συσκευή σε σύννεφο

Στο μοντέλο επικοινωνίας συσκευή σε σύννεφο, η συσκευή IoT συνδέεται απευθείας στο σύννεφο διαδικτύου. Μια υπηρεσία εφαρμογής παρέχει ανταλλαγή δεδομένων και έλεγχο κίνησης μηνυμάτων. Αυτή η προσέγγιση συχνά εκμεταλλεύεται τους μηχανισμούς επικοινωνίας του μενού, όπως οι παραδοσιακές ενσύρματες συνδέσεις Ethernet ή Wi-Fi για τη δημιουργία σύνδεσης μεταξύ συσκευής και δικτύου IP, το οποίο τελικά συνδέεται με τις υπηρεσίες cloud.

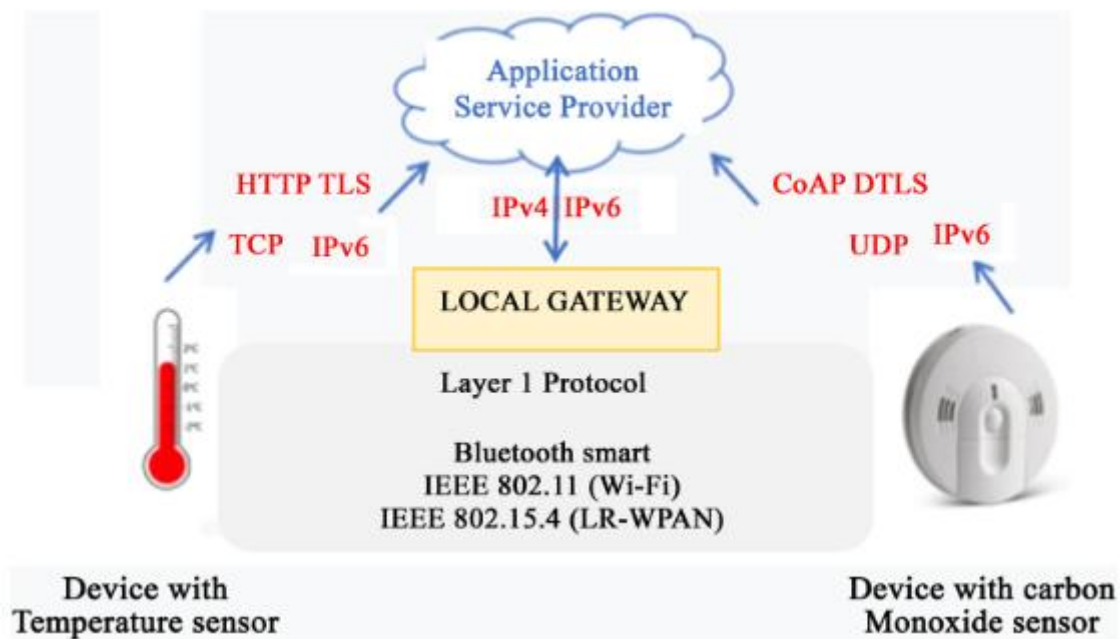


Σχήμα 27. Επικοινωνία από συσκευή σε σύννεφο

5.5.3. Επικοινωνίες συσκευής προς πύλη

Στο μοντέλο συσκευής προς πύλη, η πύλη επιπέδου συσκευής στην εφαρμογή Application Layer Gateway (ALG). Η συσκευή IoT επικοινωνεί μέσω ενός ALG που χρησιμεύει ως κανάλι πρόσβασης στις υπηρεσίες cloud. Απλώς, αυτό σημαίνει ότι υπάρχει ένα πρόγραμμα εφαρμογής που εκτελείται σε μια τοπική συσκευή πύλης που λειτουργεί ως ενδιάμεσος μεταξύ της συσκευής και της υπηρεσίας cloud και παρέχει ασφάλεια και μετάφραση δεδομένων.

Στις περισσότερες περιπτώσεις, ένα smartphone με μια εφαρμογή για επικοινωνία με μια συσκευή λειτουργεί ως τοπική πύλη και μεταφέρει τα δεδομένα σε μια υπηρεσία cloud. Συσκευές όπως ένας ιχνηλάτης γυμναστικής δεν μπορούν να συνδεθούν απευθείας στο cloud. Ως εκ τούτου, βασίζονται σε εφαρμογές έξυπνων τηλεφώνων για τη μεταφορά δεδομένων στο cloud.



Σχήμα 28. Επικοινωνία συσκευής προς πύλη

5.5.4 Πρωτόκολλα επικοινωνίας σε IoT συσκευές

Τα πρωτόκολλα επικοινωνίας στο IoT επιτρέπουν τη σύνδεση, ανταλλαγή δεδομένων και απομακρυσμένη διαχείριση έξυπνων συσκευών. Διακρίνονται σε ενσύρματα και ασύρματα, καθώς και σε πρωτόκολλα φυσικού επιπέδου, δικτύου, μεταφοράς και εφαρμογών.

Το πρωτόκολλο επικοινωνίας σε φυσικό επίπεδο που θα πρέπει να επιλεγεί για κάθε εφαρμογή εξαρτάται από την απόσταση που θα πρέπει να υπάρχει μεταξύ των συσκευών, τον όγκο των δεδομένων όπου θα ανταλλάσσουν μεταξύ τους οι συσκευές και την αυτονομία που απαιτείται σε κάθε εφαρμογή. Ο παρακάτω πίνακας είναι ενδεικτικός της ευρείας γκάμας επιλογών που έχουμε ανάλογα με την φύση της εκάστοτε εφαρμογής.

Οι υπηρεσίες Διαδίκτυο των πραγμάτων (Internet of Things) και σύγχρονες τεχνολογικές λύσεις που εφαρμόζονται στο χώρο της Υγείας

Πρωτόκολλο	Εμβέλεια	Ταχύτητα	Κατανάλωση	Χρήση
Wi-Fi (802.11)	50-100m	Έως 9.6 Gbps (Wi-Fi 6)	Μέτρια - Υψηλή	Smart home, κάμερες
Bluetooth (Classic & BLE)	10-100m	1-3 Mbps	Πολύ χαμηλή (BLE)	Wearables, smart home
Zigbee (802.15.4)	10-100m	250 kbps	Χαμηλή	Αισθητήρες, αυτοματισμοί
Z-Wave	30-100m	100 kbps	Πολύ χαμηλή	Οικιακό IoT, αυτοματισμοί
LoRaWAN	2-15 km	0.3-50 kbps	Πολύ χαμηλή	Smart cities, γεωργία
Sigfox	10-40 km	100 bps	Πολύ χαμηλή	IoT αισθητήρες μεγάλης εμβέλειας
NB-IoT (Narrowband IoT)	1-10 km	20-250 kbps	Πολύ χαμηλή	Βιομηχανία, έξυπνοι μετρητές
5G (IoT)	1-3 km	Έως 10 Gbps	Μέτρια	Βιομηχανία, αυτοκίνητα

Σχήμα 29. Ασύρματα πρωτόκολλα πρόσβασης

Ενώ τα πρωτόκολλα Bluetooth Classic και Bluetooth Low Energy έχουν πολλές ομοιότητες, όπως και τα δύο πρωτόκολλα που καλύπτονται από την προδιαγραφή Bluetooth και λειτουργούν στη ζώνη των 2,4 GHz, είναι όμως δύο διαφορετικά, ασύμβατα πρωτόκολλα.

Το Bluetooth Classic έχει σχεδιαστεί για να χειρίζεται πολλά δεδομένα, αλλά επίσης καταναλώνει γρήγορα ενέργεια. Σκεφτείτε να μεταδώσετε μουσική από το τηλέφωνό σας σε ακουστικά Bluetooth. Αυτή είναι μια παραδοσιακή περίπτωση χρήσης για το Bluetooth Classic. Το BLE, από την άλλη πλευρά, έχει σχεδιαστεί για εφαρμογές που δεν απαιτούν χειρισμό πολλών δεδομένων, αλλά απαιτούν πολύ καλή διάρκεια ζωής της μπαταρίας. Σκεφτείτε έναν αισθητήρα σε μια αποθήκη ελεγχόμενης θερμοκρασίας που θέλετε να ρυθμίσετε και να ξεχάσετε για μήνες ή και χρόνια. Το BLE θα ταίριαζε καλύτερα για τέτοιου είδους εφαρμογές. Μερικά από τα πρωτόκολλα επικοινωνίας επιπέδου εφαρμογής που χρησιμοποιούνται σε μεγάλη έκταση σε εφαρμογές αναφέρονται στον πίνακα του σχήματος 5.11.

Το CoAP είναι ιδανικό για IoT δίκτυα χαμηλής ισχύος, ενώ το MQTT είναι καλύτερο για cloud IoT επικοινωνία. Αν το IoT σύστημά μας βασίζεται σε αισθητήρες με χαμηλή κατανάλωση, το CoAP είναι η καλύτερη επιλογή.

Πρωτόκολλο	Μοντέλο Επικοινωνίας	Βασικά Χαρακτηριστικά	Χρήση
MQTT (Message Queuing Telemetry Transport)	Publish/Subscribe	Lightweight, QoS levels, TCP-based	Smart Home, Cloud IoT
CoAP (Constrained Application Protocol)	Request/Response (REST)	UDP-based, Low-power, Secure (DTLS)	Smart Cities, IoT Sensors
AMQP (Advanced Message Queuing Protocol)	Message Queue	Reliable messaging, security, transactions	Industrial IoT (IIoT)
DDS (Data Distribution Service)	Peer-to-Peer	Real-time, high-speed	Robotics, Automotive IoT
XMPP (Extensible Messaging and Presence Protocol)	Messaging	Decentralized, XML-based, Secure	IoT Chatbots, Smart Devices
HTTP/HTTPS	Request/Response	RESTful API, Web-based	Web IoT applications

Σχήμα 30. Πρωτόκολλα εφαρμογών IoT

5.5.4.1. Message Queuing Telemetry Transport (MQTT)

Το πρωτόκολλο αυτό είναι ένα πρωτόκολλο που έχει σχεδιαστεί για εφαρμογές μικρού ρυθμού μετάδοσης πληροφορίας. Βασίζεται στην λογική της εγγραφής/δημοσιοποίησης (Publish/Subscribe) ενός θέματος (topic). Υπάρχουν ενδιάμεσες συσκευές (brokers) οι οποίες είναι υπεύθυνες για να μεταδώσουν μια πληροφορία παράλληλα σε πολλές συσκευές που έχουν ζητήσει να λαμβάνουν την πληροφορία αυτή. Συγκεκριμένα ορίζεται ένα θέμα (topic) στο οποίο κάνουν οι συσκευές εγγραφή (subscribe). Οι συσκευές που έχουν την πρωτογενή πληροφορία την οποία λαμβάνουν από διάφορους αισθητήρες και έχουν κάνει επίσης εγγραφή στο συγκεκριμένο θέμα δημοσιεύουν την πληροφορία στον broker σε σχέση με αυτό το θέμα. Ο broker αφού λάβει την πληροφορία την αναμεταδίδει σε όλες τις συσκευές που έχουν ζητήσει να γίνουν αποδέκτες της πληροφορίας αυτής. Φανταστείτε για παράδειγμα ένα σύνολο αισθητήρων θερμοκρασίας όπου δημοσιεύουν περιοδικά την θερμοκρασία περιβάλλοντος. Αν κάποιος χρήστης θέλει να έχει αυτήν την πληροφορία δηλώνει το ενδιαφέρον του για αυτό το θέμα στον broker και αυτός με την σειρά του θα του δίνει την πληροφορία αυτή μόλις την λάβει.

Υπάρχουν 3 επίπεδα ποιότητας της υπηρεσίας αυτής (Quality of Service QoS)

- QoS 0 (Μέχρι μια φορά). Στέλνεται το μήνυμα και δεν υπάρχει καμία επιβεβαίωση λήψης του μηνύματος αυτού
- QoS 1 (Τουλάχιστον μια φορά). Διασφαλίζει ότι το μήνυμα θα φτάσει τουλάχιστον μια φορά αγνοώντας την περίπτωση πολλαπλών ίδιων μηνυμάτων τα οποία μπορεί να ληφθούν
- QoS 2 (Μόνο μια φορά). Εδώ διασφαλίζεται ότι θα φτάσει το μήνυμα μια μόνο φορά.

Ο broker διατηρεί το τελευταίο μήνυμα που έχει λάβει και το μεταδίδει αμέσως μόλις εντοπιστεί νέα συσκευή που έχει κάνει εγγραφή στο συγκεκριμένο topic. Από θέματα ασφαλείας υπάρχει δυνατότητα αποστολής των μηνυμάτων με TLS/SSL κρυπτογράφηση και ταυτοποίησης με χρήση username και password.

5.5.4.2. Constrained Application Protocol (CoAP)

Το CoAP (Constrained Application Protocol) είναι ένα ελαφρύ πρωτόκολλο επικοινωνίας σχεδιασμένο για συστήματα IoT με περιορισμένους πόρους, όπως αισθητήρες και μικροελεγκτές. Χρησιμοποιεί το UDP αντί για το TCP, επιτρέποντας γρήγορη και ενεργειακά αποδοτική επικοινωνία.

Υλοποιείται με RESTful API μέσω HTTP με το κλασσικό μοντέλο request/response και χρησιμοποιεί GET, POST, PUT, DELETE εντολές έναντι του Publish/Subscribe. Στο επίπεδο μεταφοράς χρησιμοποιεί το πρωτόκολλο UDP για γρηγορότερη επικοινωνία. Μπορεί να στέλνει δεδομένα ταυτόχρονα σε πολλές συσκευές. Οι συσκευές IoT μπορούν να λειτουργήσουν είτε σαν client είτε σαν servers.

Χαρακτηριστικό	CoAP	MQTT
Μοντέλο	Request/Response (REST)	Publish/Subscribe
Transport	UDP	TCP
Κατανάλωση Ενέργειας	Πολύ χαμηλή	Χαμηλή
QoS (Quality of Service)	Όχι	Ναι (QoS 0, 1, 2)
Ασφάλεια	DTLS	TLS
Χρήση	IoT αισθητήρες, Smart Cities	Cloud IoT, Smart Home

Σχήμα 31. Σύγκριση CoAP με MQTT

5.5.4.3. Advanced Message Queuing Protocol (AMQP)

Το AMQP (Advanced Message Queuing Protocol) είναι ένα πρωτόκολλο επικοινωνίας όπου βασίζεται σε ανταλλαγή μηνυμάτων και σχεδιάστηκε για αξιόπιστη και ασφαλή ανταλλαγή μηνυμάτων μεταξύ συστημάτων. Χρησιμοποιείται κυρίως σε βιομηχανικές IoT εφαρμογές, χρηματοοικονομικές συναλλαγές και cloud-based messaging.

Διασφαλίζει αξιόπιστη μεταφορά δεδομένων χωρίς απώλειες. Χρησιμοποιεί ουρές για την αποθήκευση των μηνυμάτων και επιτρέπει πολύπλοκες δρομολογήσεις των μηνυμάτων με εγγυημένη παράδοση. Τέλος είναι ανεξάρτητο από την γλώσσα προγραμματισμού ή το λειτουργικό σύστημα.

Το AMQP βασίζεται στο μοντέλο Producer – Broker – Consumer. Ο Producer δεν διανέμει τα μηνύματα απ' ευθείας στον Consumer, αλλά τα στέλνει στον Broker, ο οποίος με την σειρά του τα διαβιβάζει στις αντίστοιχες ουρές ανάλογα με τους κανόνες δρομολόγησης. Κατόπιν ο Consumer διαβάζει τα μηνύματα από την ουρά και τα επεξεργάζεται.

Χρησιμοποιείται κυρίως σε βιομηχανικές IoT εφαρμογές Industrial IoT για επικοινωνία μεταξύ μηχανών σε εργοστάσια, σε χρηματοοικονομικές συναλλαγές για διαχείριση μηνυμάτων σε τραπεζικά και ασφαλιστικά συστήματα και σε συστήματα εφοδιαστικής αλυσίδας για παρακολούθηση αποστολών και αποθεμάτων.

5.5.4.4. Data Distribution Service (DDS)

Το DDS (Data Distribution Service) είναι ένα πραγματικού χρόνου πρωτόκολλο επικοινωνίας σχεδιασμένο για υψηλής απόδοσης, χαμηλής καθυστέρησης και αξιοπιστίας εφαρμογές IoT.

Χρησιμοποιείται σε περιβάλλοντα όπου απαιτείται άμεση ανταλλαγή δεδομένων μεταξύ συσκευών, όπως αυτόνομα οχήματα, ρομποτική, αεροδιαστημική και βιομηχανικό IoT (IIoT). Εδώ η επικοινωνία δεν απαιτεί ενδιάμεση συσκευή, μειώνοντας έτσι την καθυστέρηση. Χρησιμοποιείται για εφαρμογές που απαιτούν την ελάχιστη δυνατή καθυστέρηση. Οι συσκευές μπορούν να βρίσκονται και να επικοινωνούν αυτόματα μεταξύ τους. Υπάρχουν πάνω από 15 επίπεδα ποιότητας της υπηρεσίας QoS για αξιόπιστη επικοινωνία. Εφαρμόζει την ίδια λογική με το MQTT ως προς την εγγραφή της συσκευής σε κάποιο topic και την δημοσιοποίηση στο topic αυτό, όμως για να έχει την μικρότερη δυνατή απόκριση δεν χρησιμοποιεί τον ενδιάμεσο broker αλλά οι συσκευές μιλάνε απ' ευθείας μεταξύ τους. Το πρωτόκολλο αυτό βρίσκει εφαρμογή στον τομέα των αυτόνομων οχημάτων με ανταλλαγή δεδομένων μεταξύ αισθητήρων και AI, στον τομέα της ρομποτικής και των drones με ανταλλαγή δεδομένων μεταξύ ρομπότ και ελεγκτών, στον τομέα της αεροδιαστημικής με την χρήση σε πολεμικά αεροσκάφη και σε δορυφορικά συστήματα και στον τομέα του βιομηχανικού εξοπλισμού για την παρακολούθηση και τον έλεγχο βιομηχανικού εξοπλισμού.

5.5.4.5. Extensible Messaging and Presence Protocol (XMPP)

Το XMPP (Extensible Messaging and Presence Protocol) είναι ένα ανοιχτό πρωτόκολλο επικοινωνίας βασισμένο σε XML, που σχεδιάστηκε αρχικά για άμεση ανταλλαγή μηνυμάτων, αλλά έχει επεκταθεί για χρήση σε IoT, VoIP, gaming, και σε εφαρμογές πραγματικού χρόνου.

5.5.4.6 HTTP/HTTPS Protocol

Το HTTP (HyperText Transfer Protocol) και το HTTPS (HyperText Transfer Protocol Secure) είναι τα βασικά πρωτόκολλα που χρησιμοποιούνται για τη μεταφορά δεδομένων στο διαδίκτυο. Χρησιμοποιείται για τη μεταφορά δεδομένων μεταξύ ενός web browser και ενός web server. Είναι stateless, πράγμα που σημαίνει ότι κάθε αίτημα-απόκριση είναι ανεξάρτητο από τα προηγούμενα. Ουσιαστικά υπακούει και αυτό στην client/server λογική. Ο client στέλνει ένα HTTP request και ο server απαντάει με τον αντίστοιχο HTTP response. Η απόκριση είναι σε μορφή κειμένου HTML ή JSON και είναι κρυπτογραφημένη αν το πρωτόκολλο είναι ασφαλές HTTP Secure. Οι μέθοδοι που μπορούν να χρησιμοποιηθούν είναι οι GET, POST, PUT, DELETE, PATCH και HEAD.

Μέθοδος	Περιγραφή
GET	Ζητά δεδομένα από τον server (χωρίς τροποποίηση).
POST	Στέλνει δεδομένα στον server (π.χ. φόρμες).
PUT	Ενημερώνει ένα υπάρχον πόρο ή τον δημιουργεί αν δεν υπάρχει.
DELETE	Διαγράφει έναν πόρο στον server.
PATCH	Ενημερώνει μερικώς έναν πόρο.
HEAD	Παρόμοιο με GET, αλλά επιστρέφει μόνο τις κεφαλίδες.



Σχήμα 32. HTTP μέθοδοι

Το REST (Representational State Transfer) είναι ένα αρχιτεκτονικό στυλ για σχεδιασμό web υπηρεσιών. Χρησιμοποιείται κυρίως σε APIs (RESTful APIs) που επιτρέπουν την επικοινωνία μεταξύ client και server μέσω του HTTP/HTTPS.

Με το REST είναι εφικτή η επικοινωνία μεταξύ IoT συσκευών και cloud APIs. Έξυπνες συσκευές στέλνουν δεδομένα σε cloud μέσω RESTful APIs (π.χ. θερμοκρασία αισθητήρα). Οι web εφαρμογές λαμβάνουν αυτά τα δεδομένα και τα αποτυπώνουν σε γραφικό περιβάλλον σε υπολογιστές ή σε κινητά τηλέφωνα. Είναι η πιο διαδεδομένη προσέγγιση για web services λόγω απλότητας και ευελιξίας.

5.6. Εφαρμογές IoT

Αυτό το νέο κύμα τεχνολογίας θα βρίσκεται στην ηγετική θέση για όλες τις τεχνολογίες σε όλο τον κόσμο, οι οποίες απευθύνονται σε δισεκατομμύρια συνδεδεμένες έξυπνες συσκευές που χρησιμοποιούν όλα τα δεδομένα στη ζωή μας. Με νέα ασύρματα δίκτυα, υψηλούς αισθητήρες και ανώτερες δυνατότητες, οι εφαρμογές IoT υπόσχονται να κάνουν τη ζωή μας ευκολότερη και να προσφέρουν τεράστια αξία. Ορισμένες χρήσεις των εφαρμογών IoT βρίσκονται σε αρκετούς σημαντικούς τομείς.

5.6.1 Μεταποίηση/Βιομηχανία

Η περιοχή βιομηχανικών εφαρμογών IoT καλύπτει ένα ευρύ φάσμα σύνδεσης αντικειμένων με αντικείμενα. Στο εσωτερικό, πολλά έργα εργοστασιακού αυτοματισμού και ελέγχου που βασίζονται στο IoT περιλαμβάνουν ολοκληρωμένες έξυπνες εργοστασιακές λύσεις με πολλά στοιχεία όπως παρακολούθηση ορόφου παραγωγής, απομακρυσμένο PLC ή αυτοματοποιημένα συστήματα ποιοτικού ελέγχου. Τα τυπικά έργα εκτός εργοστασίου περιλαμβάνουν τον τηλεχειρισμό των συνδεδεμένων μηχανών και την παρακολούθηση του εξοπλισμού. Αρκετές περιπτώσιολογικές μελέτες υποδεικνύουν ότι οι κύριοι μοχλοί για την παροχή λύσεων IoT είναι «ο μειωμένος χρόνος διακοπής λειτουργίας και η εξοικονόμηση κόστους».

Αυτόνομα οχήματα, ρομπότ ελεύθερης περιαγωγής κινούνται στους ορόφους των εργοστασίων. Σήμερα με τη σύγκλιση τεχνολογιών όπως η ρομποτική, οι αισθητήρες, οι τρισδιάστατες κάμερες, η συνδεσιμότητα 5G, το λογισμικό και η τεχνητή νοημοσύνη, σμήνη αυτόνομων οχημάτων έχουν βρει τον δρόμο τους με ασφάλεια στα πατώματα των εργοστασίων ως μέσο αύξησης της ταχύτητας και της ακρίβειας της συνήθους λειτουργίας. Αυτά τα ρομπότ ελεύθερης περιαγωγής μπορούν να συντονίζονται σε μεγαλύτερο βαθμό από ποτέ, επιτρέποντάς τους να εκτελούν αυτοματοποιημένες εργασίες με ελεγχόμενο και προβλέψιμο τρόπο και με ελάχιστη ανθρώπινη επίβλεψη. Αυτό τους δίνει τη δυνατότητα να βελτιώσουν τις λειτουργίες εντός των εργοστασίων παραγωγής, ειδικά σε τομείς όπως ο χειρισμός και η μεταφορά εξαρτημάτων, προσφέροντας ευκαιρίες για αύξηση της παραγωγικότητας, μείωση του κινδύνου, μείωση του κόστους και βελτίωση της συλλογής δεδομένων. Αυτό απελευθερώνει τους εργαζόμενους στο να εστιάσουν στις δραστηριότητες υψηλότερης αξίας όπως η παραγωγή και η συναρμολόγηση.

5.6.2. Μεταφορές

Διατήρηση της υγείας του οχήματος: Η τεχνολογία πρόβλεψης συντήρησης βασίζεται στη χρήση εργαλείων επικοινωνίας Internet of Things (IoT) που συλλέγουν δεδομένα σχετικά με την απόδοση διαφορετικών εξαρτημάτων, μεταφέρουν αυτά τα δεδομένα στο cloud σε πραγματικό χρόνο και αξιολογούν τον κίνδυνο πιθανής δυσλειτουργίας του υλικού ή του λογισμικού του οχήματος.

Μετά την επεξεργασία των πληροφοριών, ο οδηγός ειδοποιείται και ενημερώνεται για μελλοντική συντήρηση ή επισκευή που απαιτείται για την αποφυγή πιθανών ατυχημάτων. Με τα εργαλεία συνδεσιμότητας Internet of Things, μπορείτε να ξεχάσετε μη προγραμματισμένες στάσεις ή βλάβες κατά τη διάρκεια της διαδρομής.

Επίσης με την χρήση συστημάτων GPS μπορεί η εταιρεία μεταφορών να εντοπίζει σε πραγματικό χρόνο την πορεία της διαδρομής του οχήματος ή και του εκάστοτε θέματος και να ενημερώνει μέσω εφαρμογής τον παραλήπτη.

5.6.3 Ενέργεια

Με την παγκόσμια κατανάλωση ενέργειας να αναμένεται να αυξηθεί κατά 40% τα επόμενα 25 χρόνια, η ανάγκη για μια πιο έξυπνη ενεργειακή λύση έχει φτάσει στο υψηλότερο επίπεδο όλων των εποχών. Ευτυχώς, υπάρχουν ορισμένες σημαντικές αλλαγές προς την πιο αποτελεσματική διαχείριση ενέργειας από έξυπνους λαμπτήρες σε πλήρως αυτόνομες υπεράκτιες πλατφόρμες πετρελαίου. Συνολικά, το IoT φέρνει επανάσταση σχεδόν σε κάθε τμήμα της ενεργειακής βιομηχανίας από γενιά σε γενιά, μετάδοση στη διανομή και αλλαγή του τρόπου αλληλεπίδρασης των εταιρειών ενέργειας και των πελατών. Είναι δύσκολο να υποτιμηθεί ο σημερινός αντίκτυπος του Διαδικτύου των Πραγμάτων στον ενεργειακό τομέα. Με την αυξανόμενη ζήτηση για αυτοματοποίηση διαδικασιών και λειτουργική αποτελεσματικότητα, περισσότερες εταιρείες διερευνούν περιπτώσεις χρήσης IoT σε διαχείριση ενέργειας.

Παρακολούθηση και συντήρηση ενεργειακού συστήματος: Το IoT μπορεί να χρησιμοποιηθεί στην ενεργειακή βιομηχανία για την παρακολούθηση ορισμένων μετρήσεων του συστήματος, συμπεριλαμβανομένης της συνολικής υγείας, απόδοσης και αποδοτικότητας. Ως αποτέλεσμα, η συντήρησή τους απλοποιείται. Είτε πρόκειται για ανεμογεννήτρια, ηλιακούς συλλέκτες ή άλλο σημαντικό εξοπλισμό, μπορεί να είναι δύσκολο να εντοπιστεί ένα πρόβλημα πριν καταρρεύσει το σύστημα. Επιπλέον, ο χειροκίνητος έλεγχος για προβλήματα είναι μια πολύ σπάταλη και κουραστική διαδικασία.

Αυξημένη απόδοση: Η βελτίωση της απόδοσης των μονάδων με καύση άνθρακα είναι μία από τις πιο δύσκολες προκλήσεις στη βιομηχανία ηλεκτρικής ενέργειας σήμερα. Η τεχνολογία των εγκαταστάσεων είναι συνήθως πιο ώριμη, τα συστήματα είναι πολύ περίπλοκα και τα μέσα ποσοστά απόδοσης είναι χαμηλά. Η General Electric κυκλοφόρησε σήμερα ένα νέο πρόγραμμα ψηφιακής ενέργειας που έχει σχεδιαστεί για να διαδραματίσει σημαντικό ρόλο βοηθώντας τις χώρες να επιτύχουν τους στόχους εκπομπών αερίων του θερμοκηπίου COP21. Η χρήση του IoT συμβάλλει στην αύξηση της απόδοσης μιας μονάδας παραγωγής ενέργειας από άνθρακα στο 16%, ενώ μειώνονται οι εκπομπές αερίων του θερμοκηπίου κατά 3%. Αυτό επιτυγχάνεται με τη βελτίωση της καύσης του καυσίμου και την προσαρμογή της διαδικασίας σύμφωνα με τα χαρακτηριστικά του καυσίμου που καίγεται.

Ασφάλεια και πρόληψη καταστροφών: Οι λύσεις IoT μπορούν επίσης να χρησιμοποιηθούν στην ενεργειακή βιομηχανία για τη βελτίωση της λειτουργικής ασφάλειας και την πρόληψη ατυχημάτων στην παραγωγή, καθώς και την εξάλειψη των συνεπειών τους, όπως τα drones ασφαλείας που μπορούν να χρησιμοποιηθούν ως μέρος ενός συστήματος διαχείρισης κινδύνου

για τη μείωση των κινδύνων των εργαζομένων σε πυρηνικά εργοστάσια ή σε εγκαταστάσεις εξόρυξης.

Έξυπνοι μετρητές: Αυτές οι συσκευές ισχύος IoT συνδέουν τους καταναλωτές απευθείας στον σταθμό διανομής ηλεκτρικής ενέργειας, επιτρέποντας αμφίδρομη επικοινωνία. Έτσι, μπορούν να στείλουν κρίσιμες πληροφορίες λειτουργίας σε φορείς κοινής ωφέλειας σε πραγματικό χρόνο. Αυτό βοηθά τις υπηρεσίες κοινής ωφέλειας να αντιμετωπίζουν γρήγορα τυχόν προβλήματα απόδοσης, συμπεριλαμβανομένων των διακοπών λειτουργίας και τη μείωση του χρόνου διακοπής λειτουργίας του συστήματος. Οι έξυπνοι μετρητές μπορούν αυτόματα να εντοπίσουν και να απομονώσουν το κατεστραμμένο τμήμα της γραμμής χωρίς να επηρεαστεί η απόδοση του υπόλοιπου δικτύου. Γενικά, οι εξειδικευμένες εταιρείες διαπιστώνουν ότι υπάρχουν πολλοί τρόποι με τους οποίους οι καταναλωτές μπορούν να επωφεληθούν από τη χρήση έξυπνων μετρητών.

5.6.4. Λιανικό εμπόριο

Όλο και περισσότεροι έμποροι λιανικής συνειδητοποιούν ότι μπορούν να βελτιώσουν την αποδοτικότητα τους ως προς το κόστος και την εμπειρία των πελατών τους στο κατάστημα μέσω καινοτόμων περιπτώσεων χρήσης IoT. Υπάρχει μια τάση για τους λιανοπωλητές να ψηφιοποιήσουν τα καταστήματα και να δημιουργήσουν πιο έξυπνες λειτουργίες. Το λιανικό εμπόριο αντιπροσωπεύει πλέον το 9% των επιλεγμένων έργων, από 5%. Οι τυπικές λύσεις IoT στις λιανικές πωλήσεις περιλαμβάνουν ψηφιακή σήμανση εντός καταστήματος, παρακολούθηση και αφοσίωση πελατών, έλεγχο εμπορευμάτων, διαχείριση αποθεμάτων, έξυπνα μηχανήματα αυτόματης πώλησης και άλλα.

5.6.5 Έξυπνες πόλεις

Χάρη στη δύναμη του Διαδικτύου των Πραγμάτων, ολόκληρες πόλεις γίνονται ψηφιακά διασυνδεδεμένες και επομένως πιο έξυπνες. Συλλέγοντας και αναλύοντας τεράστιες ποσότητες δεδομένων από συσκευές IoT σε διαφορετικά συστήματα πόλεων και οι πόλεις βελτιώνουν τη ζωή των πολιτών. Οι έξυπνες πόλεις μπορούν να λάβουν καλύτερες αποφάσεις μέσω των δεδομένων που συλλέγουν για τις ανάγκες υποδομής, τις απαιτήσεις μεταφοράς, την εγκληματικότητα και την ασφάλεια. Μια μελέτη δείχνει ότι χρησιμοποιώντας υπάρχουσες εφαρμογές έξυπνων πόλεων, οι πόλεις βελτιώνουν την ποιότητα ζωής σε δείκτες (όπως η εγκληματικότητα, η κυκλοφορία και η ρύπανση) μεταξύ 10% και 30%. Οι τεχνολογίες Διαδικτύου των Πραγμάτων στην καθημερινή ζωή ως μέρος του σπιτιού, της μεταφοράς ή της πόλης σας σχετίζονται με μια πιο αποτελεσματική και ευχάριστη εμπειρία ζωής.

5.6.6. Εφοδιαστική Αλυσίδα

Καθώς οι αλυσίδες εφοδιασμού επεκτείνονται όλο και περισσότερο στους τελικούς πελάτες, με αποτέλεσμα πιο περίπλοκες ροές αγαθών των οποίων η παράδοση είναι πιο περίπλοκη, οι πάροχοι υπηρεσιών logistics ενσωματώνουν όλο και περισσότερο συνδεδεμένες ψηφιακές λύσεις για την αντιμετώπιση της πολυπλοκότητας. Στην αλυσίδα εφοδιασμού, οι συσκευές Διαδικτύου των πραγμάτων είναι ένας αποτελεσματικός τρόπος παρακολούθησης και ελέγχου ταυτότητας προϊόντων και αποστολών χρησιμοποιώντας GPS και άλλες τεχνολογίες. Επιπλέον, μπορούν επίσης να παρακολουθούν τις συνθήκες αποθήκευσης του προϊόντος που συνεπάγονται για τη διαχείριση της ποιότητας σε όλη την αλυσίδα εφοδιασμού. Οι συσκευές IoT έφεραν επανάσταση στη διαχείριση της εφοδιαστικής αλυσίδας. Είναι ευκολότερο να

καταλάβουμε πού βρίσκονται τα εμπορεύματα και πώς αποθηκεύονται επίσης όταν μπορούν να αναμένονται σε μια συγκεκριμένη τοποθεσία.

Ακολουθούν τα κύρια πλεονεκτήματα:

- 1) Επαλήθευση της θέσης των αγαθών ανά πάσα στιγμή.
- 2) Παρακολούθηση της ταχύτητας της κίνησης και πρόβλεψη χρόνου παράδοσης των αγαθών.
- 3) Βελτίωση της προβληματικής κίνησης εμπορευμάτων.
- 4) Διαχείριση των αγαθών αμέσως μετά την παραλαβή.

5.6.7 Γεωργία

Ο σημερινός παγκόσμιος πληθυσμός είναι 7,8 δισεκατομμύρια το 2020 και αναμένεται να φτάσει τα 8,6 δισεκατομμύρια το 2030, τα 10 δισεκατομμύρια το 2050 και τα 11,2 δισεκατομμύρια το 2100, σύμφωνα με τις πιο πρόσφατες εκτιμήσεις των Ηνωμένων Εθνών. Φανταστείτε πώς μπορείτε να ταΐσετε έναν τόσο τεράστιο πληθυσμό με μια απλή γεωργία. Έτσι, για να τραφεί αυτός ο τεράστιος πληθυσμός απαιτείται να παντρέψει τη γεωργία με την τεχνολογία και να έχουμε τα επιθυμητά αποτελέσματα.

Ένα έξυπνο θερμοκήπιο είναι μία από τις πολλές δυνατότητες που υπάρχουν για την επίλυση αυτού του προβλήματος. Το έξυπνο θερμοκήπιο είναι μια επανάσταση στη γεωργία, δημιουργώντας ένα αυτορρυθμιζόμενο μικροκλίμα κατάλληλο για την ανάπτυξη των φυτών μέσω της χρήσης αισθητήρων, κινητήρων και συστημάτων παρακολούθησης και ελέγχου που βελτιώνουν τις συνθήκες καλλιέργειας και αυτοματοποιούν τη διαδικασία ανάπτυξης της καλλιέργειας.

5.6.8 Υγειονομική περίθαλψη

Το Διαδίκτυο των Πραγμάτων έχει εξαπλωθεί σιγά σιγά και στον τομέα της υγείας. Ωστόσο, τα πράγματα φαίνεται να αλλάζουν υπό το πρίσμα του επίκεντρου της πανδημίας COVID-19. Τα πρώτα δεδομένα δείχνουν ότι οι ψηφιακές λύσεις υγείας που σχετίζονται με τον COVID-19 βρίσκονται σε άνοδο.

Η ζήτηση αυξάνεται για συγκεκριμένες εφαρμογές υγείας του IoT, όπως η παροχή συμβουλών τηλε-υγείας, τα ψηφιακά διαγνωστικά, η απομακρυσμένη παρακολούθηση και η ρομποτική βοήθεια. Οι εφαρμογές υγείας βοηθούν με μειωμένους χρόνους αναμονής για το δωμάτιο έκτακτης ανάγκης. Παρακολούθηση ασθενών, εργαζομένων και αποθήκης ιατροφαρμακευτικής περίθαλψης. Ενίσχυση της διαχείρισης των φαρμάκων. Εξασφάλιση της διαθεσιμότητας σημαντικών συσκευών. Το IoT έχει επίσης εισαγάγει πολλά wearable και συσκευές που έχουν κάνει τη ζωή ασθενών και γιατρών άνετη.

5.6.9 Σύγχρονες τεχνολογικές λύσεις που εφαρμόζονται στον Τομέα της Υγείας

Γενικότερα η τεχνολογία του Διαδικτύου των Πραγμάτων είναι καθοριστική στον τομέα της Υγείας καθώς προστέθηκε ένα ακόμη όπλο στην φαρέτρα της ιατρικής κοινότητας όπου ήρθε για να μείνει και να αλλάξει τον τρόπο που αντιμετωπίζουμε τις ασθένειες κυρίως σε θέματα διάγνωσης και παρακολούθησης.

5.6.9.1 Απομακρυσμένη Παρακολούθηση Ασθενούς

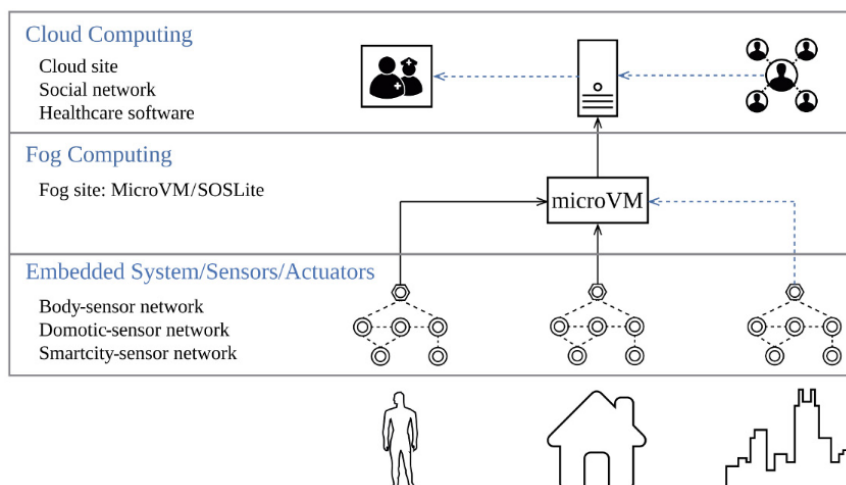
Η απομακρυσμένη παρακολούθηση ασθενών (Remote Patient Monitoring RPM) είναι μια τεχνολογία που επιτρέπει την παρακολούθηση ασθενών εκτός συμβατικών κλινικών πλαισίων, όπως στο σπίτι ή σε απομακρυσμένη περιοχή, η οποία μπορεί να αυξήσει την πρόσβαση στην περίθαλψη και να μειώσει το κόστος παροχής υγειονομικής περίθαλψης. Το RPM περιλαμβάνει τη συνεχή εξ αποστάσεως φροντίδα των ασθενών από τους ιατρούς τους, συχνά για την παρακολούθηση σωματικών συμπτωμάτων, χρόνιων παθήσεων ή αποκατάστασης μετά τη νοσηλεία. [14]

Αυτό επιτυγχάνεται με διάφορες έξυπνες συσκευές οι οποίες μπορούν να μετρούν σε καθημερινή βάση και σε μεγάλη συχνότητα για παράδειγμα την αρτηριακή πίεση σε έναν υπέρτασικό, τα επίπεδα οξυγόνου σε COVID ασθενή ή τα επίπεδα της γλυκόζης σε έναν διαβητικό ασθενή. Αυτές οι συσκευές συλλέγουν αυτές τις πληροφορίες και τις μεταδίδουν μέσω διαδικτύου σε ένα πληροφοριακό σύστημα στο οποίο έχει πρόσβαση ο ίδιος ο ασθενής ή ο θεράπωντας ιατρός σε πραγματικό χρόνο. Φανταστείτε πόσο απλοποιείται η ζωή ενός ασθενή ο οποίος κατοικεί σε ένα απομακρυσμένο νησί και τον παρακολουθεί ο ιατρός του σε ένα νοσοκομείο της ηπειρωτικής Ελλάδας. Θα ήταν αδιανόητη αυτή λύση χωρίς την τεχνολογία των έξυπνων συσκευών. Τα οφέλη αυτών των συσκευών είναι:

Προσβασιμότητα: Τόσο ο ασθενής όσο και οι πάροχοι ιατρικής φροντίδας μπορούν να έχουν πρόσβαση στα στοιχεία που συλλέγουν οι έξυπνες συσκευές οποιαδήποτε στιγμή και από οποιοδήποτε σημείο. Η πρόσβαση αυτή βοηθάει στην αποτύπωση της ακριβής εικόνας και κατά συνέπεια στην λήψη ορθών αποφάσεων.

Καλύτερος λόγος οφέλους/κόστους: Το κόστος αποτελείται μόνο από το κόστος της έξυπνης συσκευής, ενώ παράλληλα το όφελος είναι πολύ μεγαλύτερο καθώς ο ασθενής έχει ουσιαστικά το διαγνωστικό κέντρο στο σπίτι του ενώ παράλληλα παρέχεται όχι απλά μια εξέταση σε μια συγκεκριμένη στιγμή αλλά πολλαπλές εξετάσεις μέσα στην ημέρα σε πραγματικό χρόνο.

Αμεσότητα και προτροπή: Τα δεδομένα συλλέγονται, επεξεργάζονται και αναλύονται άμεσα και αυτό εξοικονομεί χρόνο. Η σύνταξη αναφορών είναι πια εύκολη υπόθεση, ενώ παράλληλα υπάρχει η δυνατότητα ειδοποίησης του ιατρού όποτε κρίνεται απαραίτητο.



Σχήμα 33. Επίπεδα της Αρχιτεκτονικής Τηλεϊατρικής [15]

Καρκίνος

Η χρήση του RPM σε ασθενείς με καρκίνο έχει αποδειχθεί ότι βελτιώνει τα αποτελέσματα συνολικά, με μελέτες να δείχνουν βελτιώσεις στα ποσοστά επανεισαγωγής στο νοσοκομείο και μειωμένη χρήση πόρων υγειονομικής περίθαλψης. Αυτές οι τεχνολογίες απομακρυσμένης παρακολούθησης συμβάλλουν στη μείωση της σοβαρότητας του πόνου καθώς και στη βελτίωση της κατάθλιψης.

Το RPM έχει βελτιώσει το προσδόκιμο ζωής των ασθενών με καρκίνο έως και 20%. Οι συσκευές απομακρυσμένης παρακολούθησης ασθενών βοηθούν σε πρώιμες παρεμβάσεις, συνταγές, τροποποιήσεις χημειοθεραπείας, κ.λπ. Το RPM έχει βοηθήσει στη μείωση των επισκέψεων στα επείγοντα περιστατικά καρκίνου ή στις παρατεταμένες θεραπείες χημειοθεραπείας. Υπολογίζεται ότι το ποσοστό νοσηλείας ασθενών με RPM είναι 2,8% και χωρίς RPM είναι 13%.

COVID 19

Η χρήση της απομακρυσμένης παρακολούθησης ασθενών μειώνει τις πρόσωπο με πρόσωπο αλληλεπιδράσεις μεταξύ ιατρού και ασθενούς.

Το RPM μπορεί να παρέχει συνέχεια φροντίδας σε συμπτωματικούς ασθενείς με COVID-19 μετά την έξοδο από το νοσοκομείο και σε ασθενείς με ήπια έως μέτρια επίπεδα αποκορεσμού οξυγόνου που δεν απαιτούν νοσηλεία και σε ασθενείς με μακροχρόνια συμπτώματα COVID-19. Λόγω της φύσης της πανδημίας, το RPM είναι ένα απαραίτητο μέσο παροχής φροντίδας σε ασθενείς που διατρέχουν κίνδυνο, όπως ηλικιωμένοι ή άτομα με ανοσοκαταστολή.

Μελέτες δείχνουν ότι η χρήση RPM κατά τη διάρκεια της πανδημίας συνέβαλε στη μείωση των νοσηλείων και στη μείωση της χρήσης πόρων οξείας φροντίδας.

Χρόνια Αποφρακτική Πνευμονοπάθεια (ΧΑΠ)

Για ασθενείς με χρόνια αποφρακτική πνευμονοπάθεια, το RPM μπορεί να συμβάλει στην αύξηση της γνώσης των ασθενών, στις έγκαιρες παρεμβάσεις και στην κοινή λήψη αποφάσεων. Ωστόσο, τα στοιχεία ποικίλουν και έχουν τεθεί επίσης ζητήματα κόστους, καθώς και η μετατόπιση των ευθυνών στους ασθενείς.

Άνοια και πτώσεις

Για ασθενείς με άνοια που διατρέχουν κίνδυνο πτώσεων, η τεχνολογία RPM προάγει την ασφάλεια και αποτρέπει τη βλάβη μέσω της συνεχούς παρακολούθησης. Οι αισθητήρες RPM μπορούν να τοποθετηθούν στο άτομο ή στις βοηθητικές συσκευές κινητικότητάς του, όπως μπαστούνια και περιπατητές. Οι αισθητήρες παρακολουθούν τη θέση, το βάδισμα, τη γραμμική επιτάχυνση και τη γωνιακή ταχύτητα ενός ατόμου και χρησιμοποιούν έναν μαθηματικό αλγόριθμο για να προβλέψουν την πιθανότητα πτώσεων, να ανιχνεύσουν αλλαγές στην κίνηση και να ειδοποιήσουν τους φροντιστές εάν το άτομο έχει πέσει. Επιπλέον, οι δυνατότητες παρακολούθησης μέσω Wi-Fi, παγκόσμιου συστήματος εντοπισμού θέσης (GPS) ή ραδιοσυχνότητας επιτρέπουν στους φροντιστές να εντοπίσουν τους περιπλανώμενους ηλικιωμένους.

Διαβήτης

Η διαχείριση του διαβήτη απαιτεί έλεγχο πολλαπλών παραμέτρων: αρτηριακή πίεση, βάρος και γλυκόζη αίματος. Η παράδοση μετρήσεων γλυκόζης αίματος και αρτηριακής πίεσης σε πραγματικό χρόνο επιτρέπει άμεσες ειδοποιήσεις για τους ασθενείς και τους παρόχους υγειονομικής περίθαλψης να επέμβουν όταν χρειάζεται. Υπάρχουν στοιχεία που δείχνουν ότι η καθημερινή διαχείριση του διαβήτη που περιλαμβάνει RPM είναι εξίσου αποτελεσματική με τη συνήθη επίσκεψη στην κλινική κάθε 3 μήνες.



Σχήμα 34. Όργανο συνεχούς παρακολούθησης γλυκόζης

Στο παραπάνω σχήμα βλέπουμε ένα όργανο συνεχούς παρακολούθησης γλυκόζης (Continuous Glucose Monitor CGM) το οποίο είναι μια συσκευή που χρησιμοποιείται για την παρακολούθηση της γλυκόζης του αίματος σε συνεχή βάση αντί για την περιοδική παρακολούθηση των επιπέδων γλυκόζης παίρνοντας μια σταγόνα αίματος από το δάχτυλο. Τα CGM χρησιμοποιούνται από άτομα που αντιμετωπίζουν τον διαβήτη τους με ινσουλίνη, για παράδειγμα άτομα με διαβήτη τύπου 1, διαβήτη τύπου 2 ή άλλους τύπους διαβήτη, όπως ο διαβήτης κύησης. Ο αισθητήρας και ο πομπός είναι στερεωμένοι στον άνω βραχίονα και ο δέκτης δείχνει το τρέχον επίπεδο γλυκόζης στο αίμα και ένα γράφημα των πρόσφατων επιπέδων γλυκόζης στο αίμα. Υπάρχουν πληθώρα ασθενειών στις οποίες μπορούν να βοηθήσουν τέτοιου τύπου έξυπνες συσκευές.

Συμφορητική καρδιακή ανεπάρκεια

Μια συστηματική ανασκόπηση της βιβλιογραφίας σχετικά με την κατ' οίκον παρακολούθηση για ασθενείς με καρδιακή ανεπάρκεια δείχνει ότι το RPM βελτιώνει την ποιότητα ζωής, βελτιώνει τις σχέσεις ασθενών-παρόχου, συντομεύει τη διάρκεια παραμονής στα νοσοκομεία, μειώνει το ποσοστό θνησιμότητας και μειώνει το κόστος για το σύστημα υγειονομικής περίθαλψης.

Υπογονιμότητα

Μια πρόσφατη μελέτη μιας λύσης απομακρυσμένης παρακολούθησης ασθενών για υπογονιμότητα έδειξε ότι για ασθενείς που είχαν ελεγχθεί κατάλληλα και αναζητούσαν θεραπεία εξωσωματικής γονιμοποίησης (IVF), ένα εξάμηνο πρόγραμμα εξ αποστάσεως παρακολούθησης είχε το ίδιο ποσοστό εγκυμοσύνης με έναν κύκλο εξωσωματικής

γονιμοποίησης. Το προϊόν και η υπηρεσία απομακρυσμένης παρακολούθησης ασθενών που χρησιμοποιήθηκαν είχαν κόστος ανά ασθενή 800 \$, σε σύγκριση με το μέσο κόστος ενός κύκλου εξωσωματικής γονιμοποίησης 15.000 \$, υποδηλώνοντας μείωση 95% στο κόστος περίθαλψης για το ίδιο αποτέλεσμα.

Τηλεϊατρική στα σωφρονιστικά συστήματα

Πρόδρομος του RPM, η Φλόριντα πειραματίστηκε για πρώτη φορά με "πρωτόγονη" χρήση τηλεϊατρικής στις φυλακές της κατά την τελευταία δεκαετία του 1980. Δουλεύοντας με τους γιατρούς Oscar W. Boultinghouse και Michael J. Davis, από τις αρχές της δεκαετίας του 1990 έως το 2007, ο Glenn G. Hammack οδήγησε την ανάπτυξη ενός πρωτοποριακού προγράμματος τηλευγείας στις κρατικές φυλακές του Τέξας από το Ιατρικό Παράρτημα του Πανεπιστημίου του Τέξας. Πολλές από τις μελέτες που διεξήχθησαν για την τηλευγεία έχουν επικεντρωθεί στην αποτελεσματικότητα σε αγροτικές ή απομακρυσμένες κοινότητες, έτσι ένα άρθρο που δημοσιεύτηκε από το plos one εξέτασε μια μελέτη που επικεντρώθηκε στην αποτελεσματικότητα της τηλευγείας στο σωφρονιστικό σύστημα. Η μελέτη έδειξε θετική επίδραση της τηλευγείας στο σωφρονιστικό σύστημα. Φάνηκε επίσης σε ορισμένες περιπτώσεις να είναι εξίσου αποτελεσματικό με την κανονική συμβατική φροντίδα.

Το Υπουργείο Υγείας του Ηνωμένου Βασιλείου έκανε μια απόπειρα καταγραφής της επίδρασης της απομακρυσμένης παρακολούθησης ασθενών η οποία ξεκίνησε τον Μάιο του 2008. Είναι η μεγαλύτερη τυχαιοποιημένη δοκιμή ελέγχου τηλευγείας και τηλεφροντίδας στον κόσμο, με τη συμμετοχή 6191 ασθενών και 238 ιατρικών γιατρών σε τρεις τοποθεσίες, το Newham, το Kent και την Cornwall. Οι δοκιμές αξιολογήθηκαν από: City University London, University of Oxford, University of Manchester, Nuffield Trust, Imperial College London και London School of Economics. Με την χρήση αυτών των έξυπνων συσκευών καταγράφηκε:

- Μείωση 45% στα ποσοστά θνησιμότητας
- Μείωση 20% στις επείγουσες εισαγωγές
- Μείωση 15% στις επισκέψεις στα επείγοντα περιστατικά
- Μείωση 14% στις ημέρες νοσηλείας
- Μείωση 8% στο κόστος των τιμολογίων

Στο Ηνωμένο Βασίλειο, ο υπουργός Υπηρεσιών Φροντίδας της Κυβέρνησης, Paul Burstow, δήλωσε ότι η τηλευγεία και η τηλεφροντίδα θα επεκταθούν τα επόμενα πέντε χρόνια (2012-2017) για να φτάσουν τα τρία εκατομμύρια άτομα.

Το RPM εξαρτάται σε μεγάλο βαθμό από το κίνητρο του ατόμου να διαχειριστεί την υγεία του. Χωρίς την προθυμία του ασθενούς να συμμετάσχει ενεργά στη φροντίδα του, η εφαρμογή RPM πιθανότατα θα αποτύχει. Η μετατόπιση της λογοδοσίας που σχετίζεται με το RPM αναδεικνύει ζητήματα ευθύνης. Δεν υπάρχουν σαφείς κατευθυντήριες γραμμές σχετικά με το εάν οι κλινικοί γιατροί πρέπει να παρεμβαίνουν κάθε φορά που λαμβάνουν ειδοποίηση, ανεξάρτητα από το επείγον. Η συνεχής ροή δεδομένων ασθενών απαιτεί μια αφοσιωμένη ομάδα παρόχων υγειονομικής περίθαλψης να χειρίζονται τις πληροφορίες, γεγονός που μπορεί, στην πραγματικότητα, να αυξήσει τον φόρτο εργασίας. Επίσης απαιτείται τυποποίηση για την ανταλλαγή δεδομένων και τη διαλειτουργικότητα μεταξύ πολλαπλών στοιχείων. Επιπλέον, η ανάπτυξη RPM εξαρτάται σε μεγάλο βαθμό από μια εκτεταμένη ασύρματη τηλεπικοινωνιακή υποδομή, η οποία μπορεί να μην είναι διαθέσιμη ή εφικτή σε αγροτικές περιοχές. Δεδομένου ότι το RPM περιλαμβάνει τη μετάδοση ευαίσθητων δεδομένων ασθενών μέσω

τηλεπικοινωνιακών δικτύων, η ασφάλεια των πληροφοριών αποτελεί ανησυχία. Η συζήτηση περιβάλλει τα πιθανά ζητήματα κυβερνοασφάλειας του RPM, συμπεριλαμβανομένης της πιθανότητας hacks που θα μπορούσαν να αντλήσουν προσωπικά ιατρικά δεδομένα.

5.6.9.2 IoT και τηλεϊατρική

Υπάρχει πλέον η δυνατότητα να κάνει διάγνωση ο ιατρός εξ αποστάσεως με την χρήση IoT αισθητήρων. Με αυτόν τον τρόπο βελτιώνεται η πρόσβαση των ασθενών στην υγειονομική περίθαλψη σε απομακρυσμένες περιοχές. Η χρήση ειδικού ιατρικού και πληροφοριακού εξοπλισμού επιτρέπει σε ιατρούς να διενεργούν βασικές εξετάσεις, όπως η καταγραφή ηλεκτροκαρδιογραφήματος (ΗΚΓ) και η σπιρομέτρηση. Τα δεδομένα αυτά αποστέλλονται ηλεκτρονικά σε εξειδικευμένα κέντρα για αξιολόγηση και διάγνωση. Οι ασθενείς μεταβαίνουν σε τοπικές εγκαταστάσεις πρωτοβάθμιας υγείας και ο ιατρός μπορεί να κάνει την εξέταση και διάγνωση από ένα κεντρικό νοσοκομείο το οποίο βρίσκεται σε επικοινωνία με τα απομακρυσμένα ιατρεία. Το αποτέλεσμα από την χρήση μιας τέτοιας υπηρεσίας είναι η απαλλαγή του ασθενούς από μεγάλες μετακινήσεις ενώ παράλληλα έχει παρατηρηθεί σε κέντρα υγείας που έχει εφαρμοστεί ότι έχουν μειωθεί και οι χρόνοι αναμονής για υποβολή εξέτασης των ασθενών. [16] Τέλος οι διαγνώσεις γίνονται πιο άμεσα με όλα τα θετικά που απορρέουν από αυτή την εξέλιξη ως προς την πορεία της υγείας του ασθενούς.

5.6.9.3 Έξυπνα νοσοκομεία και IoT εξοπλισμός

Με τις έξυπνες συσκευές παρέχετε η δυνατότητα στο νοσοκομείο να βελτιστοποιήσει την διαχείριση ασθενών και εξοπλισμού. Παράλληλα βελτιώνεται και η παρακολούθηση της διαθεσιμότητας των κρεβατιών και των αποθεμάτων των φαρμάκων, κάτι που συμβάλλει στην καλύτερη διαχείριση των πόρων και τη μείωση των καθυστερήσεων στη φροντίδα των ασθενών. Παράλληλα με ενσωμάτωση και τεχνολογίας ΑΙ μπορούμε να εκτιμήσουμε την μελλοντική διαθεσιμότητα των κλινών και κατ' επέκταση να προγραμματίσουμε τις απαιτούμενες εισαγωγές. Το αποτέλεσμα είναι η μείωση του χρόνου αναμονής, η καλύτερη διαχείριση έκτακτων περιστατικών και η αποτελεσματικότερη κατανομή των κλινών ανάλογα με την σοβαρότητα των περιστατικών.

Κεφάλαιο 6

Σχεδιασμός του Πληροφοριακού Συστήματος Smart Blister

6.1. Μοντέλο Επιχειρησιακού Σεναρίου

Προκειμένου να διασφαλιστεί η φαρμακευτική συμμόρφωση των ασθενών χρησιμοποιούνται έξυπνες συσκευασίες blisters οι οποίες διαθέτουν παθητικό ηλεκτρονικό κύκλωμα τυπωμένο πάνω στο blister με αγωγίμα μελάνια ή μελάνια μεγάλης αντίστασης ενός στρώματος και μικρή συσκευή ενεργοποίησης (activator) του blister για επικοινωνία με το κεντρικό πληροφοριακό σύστημα. Την συσκευή αυτή ενεργοποιεί ο φαρμακοποιός (pharmacist) και την χρεώνει στον ασθενή (patient). Στην διαδικασία της αρχικοποίησης ο φαρμακοποιός καταχωρεί την δοσολογία, τη συχνότητα της δόσης και τη χρονική διάρκεια της θεραπείας του φαρμάκου που έχει συνταγογραφήσει ο θεράπων ιατρός στον ασθενή. Ο θεράπων ιατρός θα μπορεί να παρακολουθεί μέσω του συστήματος την συμμόρφωση του ασθενή στην φαρμακευτική θεραπεία, να καταγράφει την πορεία της υγείας του ασθενούς και τυχόν ανεπιθύμητες ενέργειες του φαρμάκου στον ασθενή, ενώ παράλληλα θα υπάρχει η δυνατότητα ενημέρωσης και κάποιου συγγενικού προσώπου του ασθενούς (relative) σε περίπτωση αμέλειας του ασθενούς στην λήψη κάποιας δόσης του φαρμάκου. Τέλος θα υπάρχει και η δυνατότητα να βλέπουν οι φαρμακευτικές εταιρείες (company) ανώνυμες αναφορές σχετικές με την φαρμακευτική συμμόρφωση των ασθενών, την αποτελεσματικότητα των φαρμάκων και το ποσοστό ανεπιθύμητων ενεργειών. Οι αναφορές αυτές θα δίνουν μόνο στατιστικά στοιχεία και όχι φυσικά προσωποποιημένα στοιχεία με σεβασμό στα προσωπικά δεδομένα των ασθενών.

Απαραίτητη προϋπόθεση για να έχουν πρόσβαση στο σύστημα ο ιατρός, οι φαρμακευτικές εταιρείες και ο φαρμακοποιός είναι η ταυτοποίηση των χρηστών (Authenticated User), να έχουν κάνει δηλαδή εγγραφή και να έχουν συνδεθεί σ' αυτό με τα διαπιστευτήρια τους.

6.1.1. Καταγραφή Δραστών

Οι εμπλεκόμενοι Δράστες (Actors) του ΠΣ είναι οι:

- Ταυτοποιημένος Χρήστης (Authenticated User)
- Ιατρός (doctor)
- Φαρμακοποιός (pharmacist)
- Ασθενής (patient)
- Συγγενικό πρόσωπο αρθενούς (relative)

- Φαρμακευτικές Εταιρείες (company)
- Blister Activator
- Ειδοποιητής (notifier)

6.1.2. Ορισμός Δραστών

Ακολουθούν πίνακες με τους Δράστες του ΠΣ που αλληλεπιδρούν με το σχεδιαζόμενο λογισμικό.

Ταυτοποιημένος Χρήστης:

Περιγραφή	<i>Είναι όλοι χρήστες που έχουν δικαίωμα αλληλεπίδρασης με το Πληροφοριακό Σύστημα</i>
Ψευδώνυμο	<i>Authenticated User</i>
Κληρονομεί	<i>Doctor, Pharmacist, Company, Patient</i>

Ιατρός:

Περιγραφή	<i>Συνταγογραφεί τις συνταγές του ασθενούς. Ελέγχει την φαρμακευτική συμμόρφωση του ασθενούς. Καταγράφει την επίδραση των φαρμάκων στον ασθενή και τυχόν ανεπιθύμητες ενέργειες αυτών.</i>
Ψευδώνυμο	<i>Doctor</i>
Κληρονομεί	<i>Κανένας</i>

Φαρμακοποιός:

Περιγραφή	<i>Χορηγεί την συσκευή ενεργοποίησης του blister στον ασθενή καταχωρώντας την σχετική χρέωση στο σύστημα. Παράλληλα την αρχικοποιεί καταχωρώντας το φάρμακο του ασθενούς, την ποσότητα των δισκίων που διαθέτει το blister, το δοσολογικό σχήμα του φαρμάκου που έχει υποδείξει ο ιατρός καθώς και το συγγενικό πρόσωπο που ενδέχεται να λαμβάνει ειδοποίηση σε περίπτωση αμέλειας του ασθενούν να λάβει το φάρμακο.</i>
Ψευδώνυμο	<i>Pharmacist</i>
Κληρονομεί	<i>Κανένας</i>

Σχεδιασμός του Πληροφοριακού Συστήματος Smart Blister

Ασθενής:

Περιγραφή	Αλλάζει την τιμή της αντίστασης του blister και στέλνεται σήμα αφαίρεσης χαπιού κατά την εξαγωγή των δισκίων από το blister. Παράλληλα λαμβάνει σχετικές ειδοποιήσεις από το πληροφοριακό σύστημα μέσω του activator, της android εφαρμογής, email ή SMS σχετικά με την δοσολογία των φαρμάκων που έχει υποδείξει ο ιατρός.
Ψευδώνυμο	Patient
Κληρονομεί	Κανένας

Συγγενικό πρόσωπο:

Περιγραφή	Λαμβάνει ειδοποιήσεις σε περίπτωση αμέλειας του ασθενούς να λάβει μέρους της φαρμακευτικής αγωγής..
Ψευδώνυμο	relative
Κληρονομεί	Κανένας

Φαρμακευτικές Εταιρείες:

Περιγραφή	Παρακολουθούν στατιστικά στοιχεία σχετικά με την φαρμακευτική συμμόρφωση των ασθενών, την αποτελεσματικότητα των φαρμάκων καθώς και τυχόν ανεπιθύμητες ενέργειες αυτών με σεβασμό στα προσωπικά δεδομένα των ασθενών.
Ψευδώνυμο	company
Κληρονομεί	Κανένας

Blister activator:

Περιγραφή	Παρατηρεί την αντίσταση του blister και στέλνει σήμα αφαίρεσης δισκίου στο ΠΣ.
Ψευδώνυμο	activator
Κληρονομεί	Κανένας

Ειδοποιητής:

Περιγραφή	<i>Ελέγχει περιοδικά τις συνταγές όλων των ασθενών και στέλνει τις ανάλογες ειδοποιήσεις στους ασθενείς και στα συγγενικά τους πρόσωπα όπως έχουν οριστεί κατά την αρχικοποίηση.</i>
Ψευδώνυμο	<i>Notifier</i>
Κληρονομεί	<i>Κανένας</i>

6.1.3. Περιγραφή Σεναρίων Χρήσης

Στην παρούσα ενότητα παρουσιάζονται τα Σενάρια Χρήσης (Use Cases) που καταγράφηκαν κατά την συλλογή των απαιτήσεων.

Διακρίναμε 11 διακριτά Σενάρια Χρήσης, τα οποία τα κατατάξαμε σε 3 Φάσεις:

- ✓ Φάση Αρχικοποίησης χρηστών (initialize - **INIT**)
- ✓ Φάση καταγραφής γεγονότων και ενημερώσεων χρηστών (record – **REC**)
- ✓ Φάση επεξεργασίας δεδομένων (statistical processing – **STAT**)

Τα σενάρια ακολουθούν συγκεκριμένη κωδικοποίηση, η οποία είναι της μορφής:

UC-<project>-<process phase>-<ID Number>

Όπου:

UC: Use Case (Σενάριο Χρήσης)

<project>: Το αναγνωριστικό του project. Στην περίπτωση του Συστήματος SMART-BLISTER ορίζεται ως “BLSTR”

<process phase>: Η φάση στην οποία βρίσκεται το σενάριο χρήσης.

<ID Number>: Ο αύξων αριθμός του σεναρίου

6.1.3.1 Σύντομη περιγραφή των φάσεων

Έχουν οριστεί τρεις φάσεις επιχειρησιακής λογικής. Η φάση την εγγραφής των χρηστών και η αρχικοποίησης των συσκευών, φάση της καταγραφής των γεγονότων και η φάση της στατιστικής επεξεργασίας των γεγονότων.

6.1.3.1.1. Φάση Αρχικοποίησης χρηστών (initialize - INIT)

Εγγράφονται στο σύστημα οι ιατροί, οι φαρμακοποιοί, οι φαρμακευτικές εταιρείες και οι ασθενείς. Επίσης αρχικοποιείται και η συσκευή ενεργοποίησης του blister όταν πρόκειται να γίνει χρήση από κάποιον ασθενή. Χρεώνεται η συσκευή ενεργοποίησης στον ασθενή και καταχωρούνται στην συσκευή αυτή πληροφορίες όπως ο αριθμός των δισκίων ανά blister ο

συνολικός αριθμός των blister που περιέχει η συσκευασία του φαρμάκου καθώς επίσης και η δοσολογία φαρμάκων που έχει υποδείξει ο ιατρός στον ασθενή. Στην φάση αυτή μπορεί να γίνει και τροποποίηση της δοσολογίας ή αλλαγή του φαρμάκου που χρησιμοποιεί ο ασθενής σύμφωνα πάντα με τις υποδείξεις του θεράποντος ιατρού.

6.1.3.1.2. Φάση καταγραφής γεγονότων και ενημερώσεων χρηστών (record – REC)

Ελέγχεται σε πραγματικό χρόνο η εσωτερική αντίσταση του blister από τον blister acivator. Σε περίπτωση που διαπιστωθεί μεταβολή της εσωτερικής αντίστασης του blister γίνεται ενημέρωση του πληροφορικού συστήματος με τον αριθμό των δισκίων που έχουν καταναλωθεί.

Επίσης ο ιατρός καταγράφει την επίδοση των φαρμάκων με βάση τους σχετικούς δείκτες που προσδιορίζουν την πορεία της υγείας του ασθενούς. Ο ιατρός καταγράφει και τυχόν ανεπιθύμητες ενέργειες του φαρμάκου.

6.1.3.1.3 Φάση επεξεργασίας δεδομένων (statistical processing – STAT)

Όταν έχει καταχωρηθεί στατιστικά αποδεκτή ποσότητα δεδομένων οι γιατροί καθώς επίσης και οι εξουσιοδοτημένες φαρμακευτικές εταιρείες μπορούν να δουν τα αποτυπωμένα στατιστικά στοιχεία σχετικά με την φαρμακευτική συμμόρφωση των ασθενών, την αποτελεσματικότητα των φαρμάκων καθώς επίσης και τα είδη και τα ποσοστά παρενεργειών που προέκυψαν κατά την χορήγηση των φαρμάκων. Οι ιατροί μπορούν να δουν τα στοιχεία σε επίπεδο ασθενή, ενώ οι φαρμακευτικές εταιρείες μπορούν να δουν μόνο συγκεντρωτικά στατιστικά στοιχεία ανά φάρμακο και μόνο αν το φάρμακο είναι της αυτής εταιρείας.

Στην φάση αυτή γίνεται και ο έλεγχος της φαρμακευτικής συμμόρφωσης των ασθενών και αποστέλλονται ειδοποιήσεις υπό μορφή ηλεκτρονικών μηνυμάτων (SMS ή emails) στους ασθενείς και τους συγγενείς τους.

6.1.4. Αναλυτική περιγραφή των σεναρίων

Παρακάτω ακολουθούν στην συνέχεια όλα τα Σενάρια Χρήσης.

6.1.4.1. UC-BLSTR-INIT-1 Εγγραφή χρηστών

Γίνεται εγγραφή των χρηστών στο ΠΣ. Κατά την διαδικασία εγγραφής καταχωρούνται διαφορετικά στοιχεία ανάλογα με τον τύπο του χρήστη. Ο χρήστης που μπορεί να καταγραφεί είναι είτε ιατρός, είτε φαρμακοποιός, είτε φαρμακευτική εταιρεία είτε ο ασθενής. Αν ο ασθενής δεν έχει την ευχέρεια στην διαχείριση του λογαριασμού του θα μπορούσε ν' αναθέσει τον ρόλο του σε κάποιο συγγενικό του πρόσωπο. Οι χρήστες έχουν υποχρεωτικά και προαιρετικά πεδία. Στην περίπτωση της φαρμακευτικής εταιρείας θα πρέπει να οριστεί ο εταιρικά υπεύθυνος του οποίου θα καταχωρηθούν τα στοιχεία επικοινωνίας του ΠΣ με την εταιρεία. Η καταχώρηση γίνεται σε δύο υποφάσεις. Στην πρώτη ο χρήστης καταχωρεί τα στοιχεία του. Γίνονται οι σχετικοί έλεγχοι και κατόπιν περνάει στην φάση της προεγγραφής. Αμέσως μετά αποστέλλετε

ένας σύνδεσμος στην διεύθυνση ηλεκτρονικού ταχυδρομείου που έχει ορίσει ο χρήστης ως διεύθυνση ηλεκτρονικής επικοινωνίας. Επιλέγοντας ο χρήστης το ληφθέν σύνδεσμο γίνεται και η οριστική ενεργοποίηση του χρήστη. Τέλος αποστέλλεται και ένα ηλεκτρονικό μήνυμα επιβεβαίωσης οριστικής εγγραφής του χρήστη.

6.1.4.2. UC-BLSTR-INIT-2 Καταχώριση δοσολογικού σχήματος

Ο ιατρός αφού συνταγογραφήσει μια συνταγή στον ασθενή καταγράφει το δοσολογικό σχήμα το οποίο θα παραλάβει ο φαρμακοποιός για να το καταχωρίσει στο σύστημα.

6.1.4.3. UC-BLSTR-INIT-3 Αρχικοποίηση συσκευής ενεργοποίησης blister

Ο φαρμακοποιός χρεώνει μία συσκευή ενεργοποίησης blister ανά είδος φαρμάκου στον ασθενή. Η χρέωση γίνεται με χρήση ενός barcode reader ή ενός QR-code reader ή ακόμη καταχωρώντας το id της συσκευής ενεργοποίησης στο ΠΣ μέσω πληκτρολογίου. Παράλληλα καταχωρεί στην συσκευή ενεργοποίησης το είδος του φαρμάκου και την δοσολογία των φαρμάκων όπως την έχει υποδείξει ο ιατρός, το email του ασθενούς και του συγγενικού του προσώπου που θα λαμβάνει την σχετική ενημέρωση. Αντί για email ο φαρμακοποιός μπορεί να καταχωρεί τον αριθμό κινητού τηλεφώνου αν έχει επιλεγεί να γίνεται η ενημέρωση μέσω SMS αντί για email. Σ' αυτό το σενάριο ο φαρμακοποιός θα πρέπει να εισέλθει στο σύστημα ως εξουσιοδοτημένος χρήστης.

6.1.4.4. UC-BLSTR-INIT-4 Μεταβολή στοιχείων συσκευής ενεργοποίησης blister

Ο φαρμακοποιός μπορεί να μεταβάλλει τα στοιχεία αρχικοποίησης της συσκευής αν έχει αλλάξει το φάρμακο, η δοσολογία η ακόμη και όταν απλά ο ασθενής χρειάζεται να αγοράσει νέο ίδιο φάρμακο λόγω κατανάλωσης των συνταγογραφηθέντων φαρμάκων και εκτέλεσης νέας συνταγής.

6.1.4.5. UC-BLSTR-INIT-5 Ορισμός και μεταβολή στοιχείων ειδοποιήσεων

Ο φαρμακοποιός μπορεί να δώσει κάποιες ρυθμίσεις που αφορούν κυρίως το email αποστολής ενημερωτικών μηνυμάτων του ασθενούς, το είδος των μηνυμάτων που θα λαμβάνει ο ασθενής, την συχνότητα αποστολής ειδοποιήσεων κ.λ.π.

6.1.4.6. UC-BLSTR-REC-1 Παρακολούθηση αντίστασης του blister

Η συσκευή ενεργοποίησης του blister παρακολουθεί την αντίσταση του blister. Μόλις αφαιρεθεί ένα δισκίο από τον ασθενή και μεταβληθεί η αντίσταση του blister εκτελείται μια ρουτίνα διακοπής και γίνεται ενημέρωση του ΠΣ σχετικά με την νέα τιμή των υπολειπόμενων δισκίων. Επίσης, καταχωρείται το συμβάν της λήψης των φαρμάκων στο ιστορικό του ασθενούς.

6.1.4.7. UC-BLSTR-REC-2 Καταχώρηση κατάστασης υγείας ασθενούς και τυχόν ανεπιθύμητων ενεργειών

Ο ιατρός αφού εισέλθει στο σύστημα ως πιστοποιημένος χρήστης θα μπορεί αφού δει τις εξετάσεις που έχει προσκομίσει στον ασθενή να καταχωρήσει κάποιους δείκτες ανάλογα με την ασθένεια και να βλέπει με αυτόν τον τρόπο την αποτελεσματικότητα του φαρμάκου. Οι δείκτες αυτοί θα υπάρχουν στο σύστημα σε προκαθορισμένες λίστες και θα παίρνουν αριθμητικές τιμές. Ο τρόπος αυτός είναι καθοριστικός για να μπορούν αργότερα να παραχθούν αξιόπιστα συμπεράσματα για την αποτελεσματικότητα των φαρμάκων. Επίσης θα μπορεί να καταχωρεί τυχόν ανεπιθύμητες ενέργειες που παρατηρήθηκαν κατά την χορήγηση των φαρμάκων.

6.1.4.7. UC-BLSTR-STAT-1 Παρακολούθηση φαρμακευτικής συμμόρφωσης

Ο ιατρός αφού εισέλθει στο σύστημα ως πιστοποιημένος χρήστης θα μπορεί να βλέπει την φαρμακευτική συμμόρφωση του ασθενούς. Έτσι σε περίπτωση που δει κάτι ανησυχητικό θα μπορεί να επικοινωνήσει με τον ασθενή.

6.1.4.8. UC-BLSTR-STAT-2 Παρακολούθηση διαθεσιμότητας φαρμάκων

Ο φαρμακοποιός αφού εισέλθει στο σύστημα ως πιστοποιημένος χρήστης θα μπορεί να βλέπει σε πραγματικό χρόνο τα υπόλοιπα φάρμακα των ασθενών του. Αυτό φυσικά δεν θα γίνεται για κάθε ασθενή ξεχωριστά καθώς κάτι τέτοιο θα ήταν ανέφικτο. Θα μπορούσε να όμως να λαμβάνει μια συγκεντρωτική κατάσταση ανά φάρμακο για φάρμακα που είναι στο όριο να τελειώσουν από τους ασθενείς.

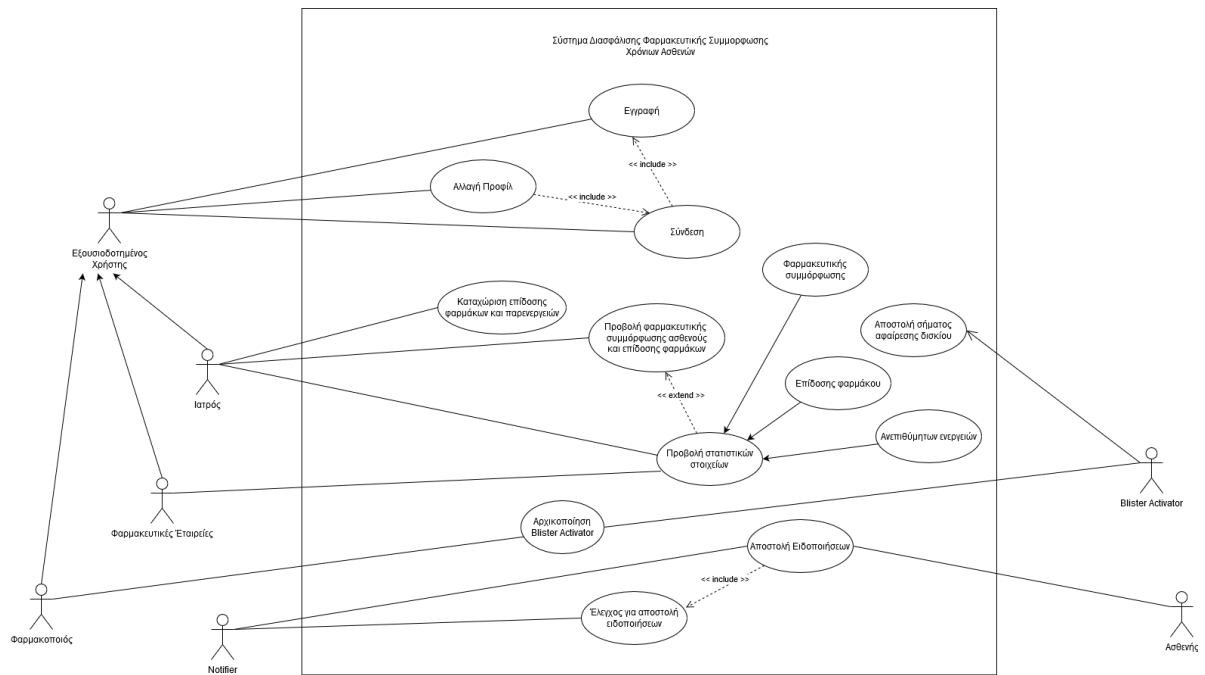
6.1.4.9. UC-BLSTR-STAT-3 Παρακολούθηση αποτελεσματικότητας φαρμάκων και παρενεργειών

Οι φαρμακευτικές εταιρείες θα μπορούν να βλέπουν συγκεντρωτικά στατιστικά στοιχεία για την αποτελεσματικότητα των φαρμάκων τους, το ποσοστό των ασθενών που διέκοψαν την θεραπεία και τις πιθανές ανεπιθύμητες ενέργειες που μπορεί να προκαλούν τα φάρμακα στους ασθενείς. Το ίδιο μπορούν να δουν και οι ιατροί αυτοί όμως θα μπορούν να δουν και σε επίπεδο ασθενούς.

6.1.4.10. UC-BLSTR-STAT-4 Αποστολή ειδοποιήσεων στον ασθενή ή και στον συγγενή του

Το ΠΣ εκτελεί μια ρουτίνα σε συχνότητα που θα ορίσει ο διαχειριστής του συστήματος με βάση τις οριζόμενες προδιαγραφές. Η συχνότητα μπορεί να είναι το ελάχιστο 5 λεπτά. Κατά την εκτέλεση αυτής της ρουτίνας θα γίνονται έλεγχοι σχετικά με τα φάρμακα που πρέπει να λάβουν οι ασθενείς και με τα μηνύματα που έχουν ορίσει οι χρήστες να λαμβάνουν. Μετά τον σχετικό έλεγχο θα στέλνονται τα σχετικά ηλεκτρονικά μηνύματα στους ασθενείς και στα συγγενικά τους πρόσωπα.

6.1.5. Διάγραμμα περίπτωσης χρήσης (use case diagram)



Σχήμα 35. Use Case Diagram

6.1.6. Διαπροσωπίες Χρηστών

6.1.6.1. Διαπροσωπία Φαρμακοποιού

Smart-Blister

Αποσύνδεση

Διεύθυνση Φαρμακείου

Τηλέφωνο Φαρμακείου

Υπεύθυνος Φαρμακείου

Αναφορές

Αρχικοποίηση Συσκευής

Προφίλ

☐ Κατάσταση φαρμάκων που προβλέπεται να ζητηθούν

☐ Κατάσταση ασθενών που διαθέτουν smart - blister

☒ Κατάσταση φαρμάκων που διαθέτουν τεχνολογία smart - blister

☐ Κατάσταση

Εκτύπωση

Σχήμα 36. Διαπροσωπία Φαρμακοποιού - Αναφορές

Smart-Blister

Αποσύνδεση

Διεύθυνση Φαρμακείου

Τηλέφωνο Φαρμακείου

Υπεύθυνος Φαρμακείου

Αναφορές

Αρχικοποίηση Συσκευής

Προφίλ

ΑΜΚΑ Ασθενούς

Όνοματεπώνυμο Ασθενούς

Τηλέφωνο Ασθενούς

Αναζήτηση Ασθενούς

Όνοματεπώνυμο θεράποντος ιατρού

Τηλέφωνο θεράποντος ιατρού

Ασθένεια

Φαρμακευτική Αγωγή

Φάρμακο 1

Φάρμακο 2

Φάρμακο 3

Εισαγωγή

Διαγραφή

Δοσολογία

Δισκία ανά δόση

Συνολικά δισκία θεραπείας

Ημ/νία Έναρξης Θεραπείας

Ημ/νία Ολοκλήρωσης Θεραπείας

ID συσκευής ενεργοποίησης

Ενεργή

Προγραμματισμός Δόσεων

09:00

21:00

Εισαγωγή

Διαγραφή

Ενημέρωση

Σχήμα 37. Διαπροσωπία Φαρμακοποιού – Αρχικοποίηση Συσκευής

Smart-Blister
Αποσύνδεση

Διεύθυνση Φαρμακείου
Τηλέφωνο Φαρμακείου

Υπεύθυνος Φαρμακείου

Αναφορές
Αρχικοποίηση συσκευής
Προφίλ

Διεύθυνση Φαρμακείου

Νομός

▼

Δήμος

▼

Οδός

Αριθμός

T.K.

ΑΦΜ

ΔΟΥ

▼

Τηλ. Σταθερό

email

Υπεύθυνος Φαρμακείου

Επώνυμο

Όνομα

Πατρώνυμο

Τηλ Κινητό

Αρ. Ταυτότητας

Είδος Ταυτότητας

▼

Ημ. Έκδοσης

Εκδούσα Αρχή

Ενημέρωση

Ακύρωση

Σχήμα 38. Διαπροσωπία Φαρμακοποιού – Αναφορές Προφίλ

6.1.6.2 Διαπροσωπία Ιατρού

Smart-Blister

Αποσύνδεση

Όνοματεπώνυμο Ιατρού

Πατρώνυμο

Ειδικότητα

Αναφορές

Επίδοση / Δοσολογία

Προφίλ

☐ Κατάσταση επιδόσεων φαρμάκων

☐ Κατάσταση ανεπιθύμητων ενεργειών φαρμάκων

☒ Κατάσταση φαρμακευτικής συμμόρφωσης ασθενών

☐ Κατάσταση

Εκτύπωση

Σχήμα 39. Διαπροσωπία Ιατρού - Αναφορές

Smart-Blister

Αποσύνδεση

Ονοματεπώνυμο Ιατρού

Πατρώνυμο

Ειδικότητα

Αναφορές

Επίδοση / Δοσολογία

Προφίλ

ΑΜΚΑ Ασθενούς

Ονοματεπώνυμο Ασθενούς

Αναζήτηση Ασθενούς

Σταθερό Τηλέφωνο Ασθενούς

Κινητό Τηλέφωνο Ασθενούς

Ασθένεια

Φαρμακευτική Αγωγή

Φάρμακο 1

Φάρμακο 2

Φάρμακο 3

Εισαγωγή

Διαγραφή

Δοσολογία

Δισκία ανά δόση

Συνολικά δισκία θεραπείας

Ημ/νία Έναρξης Θεραπείας

Ημ/νία Ολοκλ/σης Θεραπείας

Προγραμματισμός Δόσεων

09:00

21:00

Εισαγωγή

Διαγραφή

Ιστορικό Δοσολογιών

Καταχώριση δεικτών επίδοσης

Καταχώριση ανεπιθύμητων ενεργειών

Σχήμα 40. Διαπροσωπία Ιατρού – Επίδοση / Δοσολογία

75

Smart-Blister

Αποσύνδεση

Ονοματεπώνυμο Ιατρού

Πατρώνυμο

Ειδικότητα

Αναφορές

Επίδοση / Δοσολογία

Προφίλ

Όνομα

Επώνυμο

Πατρώνυμο

Μητρώνυμο

Ημ. Γέννησης

ΑΦΜ

ΔΟΥ

ΑΜΚΑ

Αρ. Ταυτότητας

Τύπος Ταυτότητας

Εκδούσα Αρχή

Ημ. Έκδοσης

Νομός Κατοικίας

Δήμος

Οδός

Αριθμός

T.K.

Τηλ. Σταθερό

Τηλ Κινητό

email

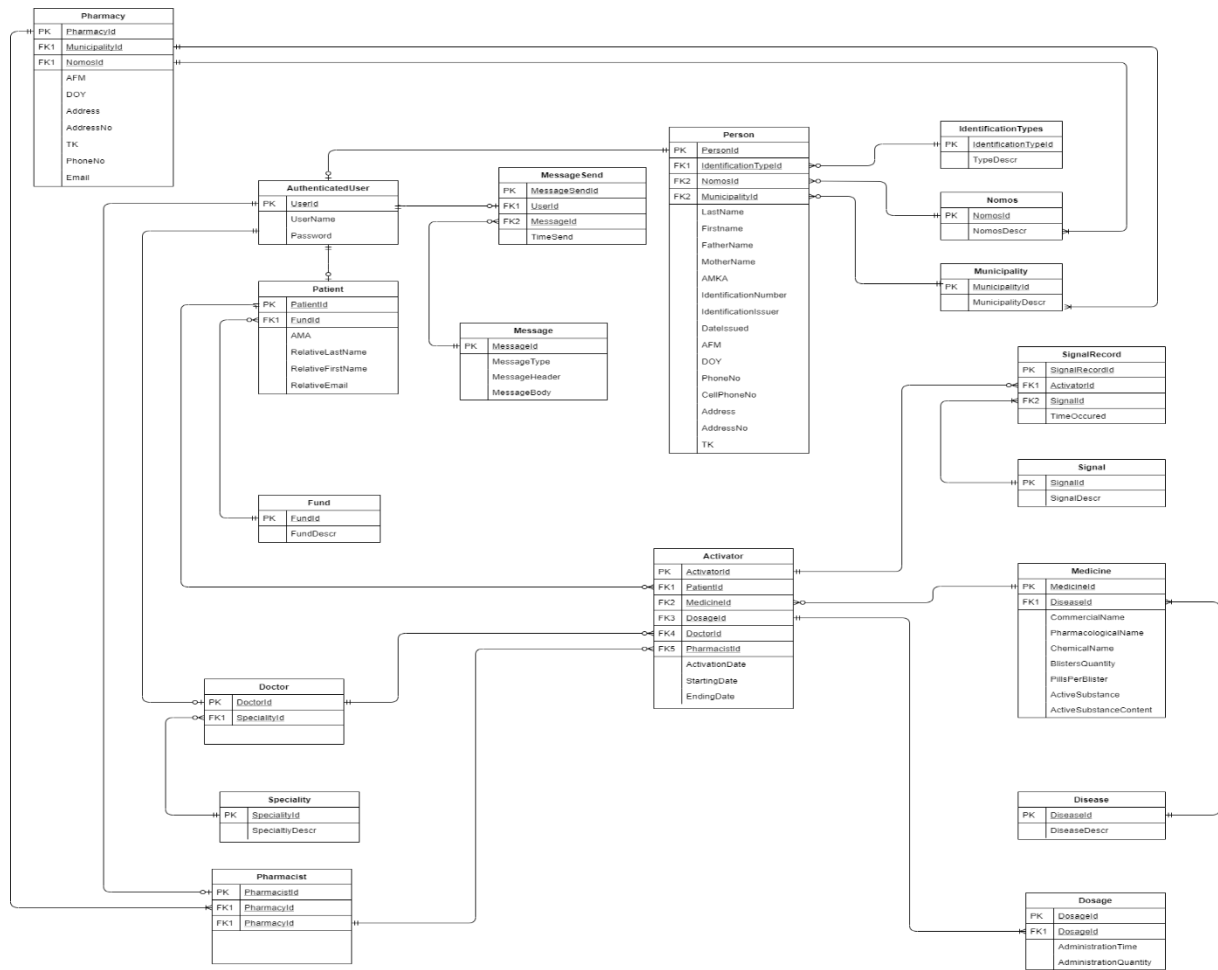
Ειδικότητα

Ενημέρωση

Άκυρο

Σχήμα 41. Διαπροσωπία Ιατρού – Προφίλ

6.1.7. Μοντέλο Οντοτήτων Συσχετίσεων (Entity Relationship Diagram)



Σχήμα 42. Μοντέλο Οντοτήτων Συσχετίσεων

6.2. Επιλογή Τεχνολογίας

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε μία εικονική μηχανή με εγκατεστημένο λειτουργικό σύστημα ubuntu έκδοσης 18.04.6 LTS. Για την μηχανή αυτή έγινε χρήση cloud υπηρεσίας IaaS από πάροχο εξοικονομώντας έτσι την υποδομή χωρίς να μπορούμε στην διαδικασία αγοράς εξοπλισμού στον οποίο θα έπρεπε να διασφαλίσουμε συνθήκες αδιάλειπτης παροχής ηλεκτρικής ενέργειας και κατάλληλης θερμοκρασίας.

Στο μηχανήμα αυτό σε επόμενη φάση εγκαταστήσαμε web server Apache 2.4.29 και DNS server για να μπορούμε να έχουμε πρόσβαση στην web υπηρεσία μέσα από το διαδίκτυο.

Για την αποδοτική υλοποίηση της εφαρμογής ο κώδικας γράφτηκε σε γλώσσα προγραμματισμού python με χρήση το Django Framework 3.0.7. Γενικά τα frameworks είναι ένα σύνολο εργαλείων, βιβλιοθηκών και κανόνων που βοηθούν τους προγραμματιστές να δημιουργούν εφαρμογές πιο γρήγορα και αποτελεσματικά. Παρέχει μια έτοιμη δομή και αυτοματοποιεί πολλές διαδικασίες, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στη λογική της εφαρμογής και όχι στη βασική υποδομή. Ειδικότερα το Django Framework παρέχει έτοιμες λύσεις για CRUD (Create, Read, Update, Delete), για αυθεντικοποίηση ενώ διαθέτει και ένα εργαλείο για τον διαχειριστή του συστήματος. Περιέχει ενσωματωμένες λειτουργίες για βάσεις δεδομένων, για σχεδιασμό περιγραμμάτων σελίδων (templates), ενώ διαθέτει και το Django REST Framework (DRF) για γρήγορη και αποδοτική ανάπτυξη RESTful APIs.

Το Django ακολουθεί το MVT σχεδιαστικό πρότυπο (Model View Template). Τα τρία αυτά επίπεδα είναι διαχωρισμένα μεταξύ τους για αποδοτικότερη και αποτελεσματικότερη υλοποίηση της εφαρμογής.

Στο model επίπεδο ορίζεται η βάση δεδομένων στην οποία θα αποθηκεύονται τα δεδομένα. Μπορεί να χρησιμοποιηθεί οποιαδήποτε από της MySQL, PostgreSQL, SQLite, Oracle ή MariaDB και μπορείς να μεταβείς από την μια βάση στην άλλη με μεγάλη ευκολία. Τα μοντέλα της βάσης αποθηκεύονται στο αρχείο models.py

Το view επίπεδο λαμβάνει τα http requests και είναι υπεύθυνο για να αναζητήσει, να αντλήσει και να επιστρέψει τα δεδομένα που ζητούνται. Ο σχετικός κώδικας αυτού του επιπέδου αποθηκεύεται στο αρχείο views.py

Τέλος το επίπεδο template είναι υπεύθυνο για τον τρόπο που θα εμφανιστούν τα δεδομένα. Τα templates είναι συνήθως .html αρχεία με τον html κώδικα να περιγράφει το περίγραμμα της σελίδας που θέλουμε να εμφανιστεί στον web browser.

Για την υλοποίηση του πηγαίου κώδικα χρησιμοποιήθηκε το σύστημα Git το οποίο είναι ένα σύστημα διαχείρισης εκδόσεων λογισμικού version control system. Με αυτό το σύστημα μπορείς να ορίζεις σημεία αναφοράς στα οποία μπορείς να επανέλθεις όποτε το επιθυμείς. Από τα σημεία αυτά μπορείς να εξελίξεις τον κώδικά σου και να δημιουργήσεις νέα σημεία αναφοράς ή να επιστρέψεις σε προηγούμενα σημεία αναφοράς και να ακολουθήσεις από εκεί άλλα βήματα υλοποίησης. Επίσης μπορείς να διαμοιράσεις τον κώδικά σου στους συνεργάτες σου ώστε να δουλεύετε παράλληλα σε διαφορετικά σημεία του κώδικα και μετά την ολοκλήρωση των επιμέρους κομματιών να γίνει η επανασύνδεση τους σε ένα ολοκληρωμένο τελικό έργο. Το GitHub διαχειρίζεται τις λειτουργίες του Git σε ένα web based περιβάλλον με αποτέλεσμα να αξιοποιούνται πιο εύκολα οι δυνατότητες συνεργασίας με άλλους, διαμοιρασμού κώδικα αλλά και αποθήκευσης των έργων στο διαδίκτυο. Ο κώδικας που έχω υλοποιήσει έχει αποθηκευτεί στο GitHub στη διεύθυνση <https://github.com/antoniodakis/smartblister>

Επίσης για την αποδοτικότερη υλοποίηση της web εφαρμογής χρησιμοποιήθηκε το framework Bootstrap. Το Bootstrap είναι ένα δημοφιλές front-end framework που χρησιμοποιείται για τη δημιουργία responsive και mobile-first ιστοσελίδων. Είναι ένα σύνολο εργαλείων που

περιλαμβάνει HTML, CSS και JavaScript και βοηθά τους προγραμματιστές να σχεδιάσουν γρήγορα και εύκολα σύγχρονες ιστοσελίδες και web εφαρμογές.

Τα κύρια χαρακτηριστικά του bootstrap είναι:

- **Responsive Design** – Οι ιστοσελίδες προσαρμόζονται αυτόματα σε διαφορετικά μεγέθη οθονών (κινητά, tablet, desktop).
- **Grid System** – Διαθέτει ένα ευέλικτο σύστημα πλεγμάτων (grid system) που επιτρέπει την εύκολη οργάνωση περιεχομένου.
- **Έτοιμα Στυλ και Components** – Περιλαμβάνει προκαθορισμένα στυλ για κουμπιά, φόρμες, κάρτες, μενού πλοήγησης και άλλα στοιχεία.
- **Υποστήριξη JavaScript Plugins** – Παρέχει έτοιμα JavaScript plugins για λειτουργίες όπως sliders, modals και dropdown menus.
- **Ευκολία Χρήσης** – Ακόμα και αρχάριοι μπορούν να δημιουργήσουν όμορφες και λειτουργικές σελίδες γρήγορα.
- **Συμβατότητα με όλους τους σύγχρονους browsers.**

Όπως αναφέραμε και προηγουμένως υλοποιήθηκε και το RESTful API μέσω του εργαλείου Django REST Framework (DRF). Μέσω αυτού το API θα μπορεί να στέλνει οποιαδήποτε Smart Blister συσκευή HTTP requests για να καταγράφονται οι κινήσεις σχετικά με λήψη χαπιών. Οι συσκευές θα πρέπει να ακολουθεί τους κανόνες του σχετικού API προκειμένου να συνεργάζεται αρμονικά με την web εφαρμογή.

Στη συγκεκριμένη εργασία υλοποιήσαμε μια εφαρμογή κινητού τηλεφώνου η οποία θα στέλνει αυτά τα HTTP requests στην web εφαρμογή και θα ενημερώνονται οι σχετικοί πίνακες στη βάση δεδομένων.

Η εφαρμογή κινητού τηλεφώνου υλοποιήθηκε μέσω του εργαλείου MIT App Inventor το οποίο είναι εργαλείο που έχει υλοποιηθεί από το Πανεπιστήμιο της Μασαχουσέτης. Το MIT App Inventor είναι ένα δωρεάν, block-based περιβάλλον προγραμματισμού που επιτρέπει τη δημιουργία Android εφαρμογών χωρίς προηγούμενη γνώση κώδικα. Είναι ιδιαίτερα χρήσιμο για IoT projects, καθώς επιτρέπει τη σύνδεση εφαρμογών με IoT συσκευές μέσω Bluetooth, Wi-Fi και APIs. Διαθέτει μια εύχρηστη διεπαφή χρήστη και ο προγραμματισμός γίνεται με drag and drop blocks. Επίσης δεν απαιτεί εγκατάσταση καθώς η ανάπτυξη των εφαρμογών γίνεται μέσω web browser ενώ σου δίνει παράλληλα και την δυνατότητα να αποθηκεύσεις τις εφαρμογές σου στο cloud. Φυσικά μπορείς πέρα την χρήση του προσομοιωτή που διαθέτει να κάνεις εγκατάσταση της εφαρμογής σου απ' ευθείας στο κινητό σου.

Κεφάλαιο 7

Επίλογος (Συμπεράσματα και ιδέες για μελλοντικές επεκτάσεις)

7.1. Συμπεράσματα

Η υλοποίηση ενός πληροφοριακού συστήματος έτοιμου να συνεργαστεί με οποιαδήποτε συσκευή η οποία θα είναι σε θέση να ακολουθήσει τις προδιαγραφές του καθορισμένου πρωτοκόλλου επικοινωνίας σε επίπεδο εφαρμογής θα μπορούσε να καταστήσει το σύστημα ιδιαίτερα χρήσιμο στην βελτίωση της φαρμακευτικής συνέπειας των ασθενών.

Το βασικό συστατικό της επιτυχίας ενός τέτοιου πληροφοριακού συστήματος είναι η απλότητα του πρωτοκόλλου επικοινωνίας σε επίπεδο εφαρμογής και η συμβατότητά του με οποιαδήποτε IoT συσκευή. Αυτό συμβαίνει διότι τα μηνύματα είναι απλά μηνύματα κειμένου σε μορφή JSON ή XML σε κρυπτογραφημένη βέβαια μορφή που καθιστούν την διασύνδεση απλή και συμβατή με οποιαδήποτε πιθανή smartblister συσκευή.

Παράλληλα πέρα από την βοήθεια που θα παρέχει στον ασθενή στο να διατηρήσει την φαρμακευτική του συμμόρφωση σε υψηλά επίπεδα θα βοηθήσει σημαντικά τόσο τον θεράποντα ιατρό που θα έχει πλήρη και ακριβή εικόνα ως προς την συνέπεια του ασθενούς και της αντίδρασης του οργανισμού του στην προτεινόμενη θεραπεία, όσο και στις φαρμακευτικές εταιρείες καθώς θα μπορούν να λάβουν ένα πολύ μεγάλο δείγμα σχετικά με την εξέλιξη της υγείας των ασθενών στο χορηγούμενο φάρμακο και για το ποσοστό των πιθανών παρενεργειών που μπορεί να προκαλεί η χορήγηση του συγκεκριμένου φαρμάκου.

7.2. Ιδέες για μελλοντικές επεκτάσεις

Στην εφαρμογή θα μπορούσαν να ενσωματωθούν αρκετές μετρικές για να γίνεται ακόμη καλύτερη η οπτικοποίηση των δεδομένων που συλλέγονται από το σύστημα. Ακόμη θα μπορούσαν να εφαρμοστούν τεχνολογίες τεχνητής νοημοσύνης ώστε να μπορούν να βγουν ακόμη περισσότερα συμπεράσματα σχετικά με τους λόγους μη συμμόρφωσης μεγάλου μέρους του πληθυσμού. Με αυτές τις τεχνολογίες ΑΙ θα μπορούσε να γίνει πιο αποτελεσματική η επικοινωνία του συστήματος με τον ασθενή και τις φαρμακευτικές εταιρείες αλλά κυρίως ο ιατρός θα μπορούσε να αποκομίσει πιο αξιόπιστα συμπεράσματα ως προς την αποτελεσματικότητα του κάθε φαρμάκου.

Άλλωστε οι cloud εφαρμογές που έχουν την δυνατότητα να αποκομίζουν τεράστιο όγκο δεδομένων σε πραγματικό χρόνο και με συνεχή ροή προσφέρουν το απαιτούμενο έδαφος για χρήση ΑΙ τεχνολογιών.

Κεφάλαιο 8

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

Ο κώδικας της εφαρμογής έχει γραφτεί σε γλώσσα προγραμματισμού python. Φυσικά επειδή ο κώδικας είναι αρκετά μεγάλος δεν έχει νόημα να αποτυπωθεί όλη η υλοποίηση στην διπλωματική εργασία. Έχουν αποτυπωθεί κάποια ενδεικτικά αρχεία κώδικα. Ωστόσο ο πλήρης κώδικας της εφαρμογής υπάρχει στο GitHub και συγκεκριμένα στη διεύθυνση <https://github.com/antonioudakis/smartblister>.

Το GitHub είναι μια διαδικτυακή πλατφόρμα αποθετηρίων κώδικα που επιτρέπει στους προγραμματιστές να αποθηκεύουν, να διαχειρίζονται και να συνεργάζονται με άλλους προγραμματιστές σε έργα λογισμικού. Συνεργάζεται με το σύστημα Git το οποίο είναι ένα σύστημα διαχείρισης εκδόσεων λογισμικού.

Όπως είχαμε αναφέρει και στο κεφάλαιο 6, το Django ακολουθεί το MVT σχεδιαστικό πρότυπο (Model View Template) με το οποίο διαχωρίζεται ο κώδικας της δημιουργίας των μοντέλων (models.py), όπου έχει να κάνει με την δημιουργία της βάσης δεδομένων, ο κώδικας της λήψης των HTTP requests και της εξόρυξης και κοινοποίησης των αντίστοιχων δεδομένων (view.py) και ο κώδικας του τρόπου εμφάνισης των δεδομένων με τα διάφορα html templates όπως είναι αρχεία html μορφής.

Smartblister urls.py

```
from django.contrib import admin
from django.urls import path, include
from django.contrib.auth import views as auth_views
from dashboard import views as dashboard_views
from users import views as users_views

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", dashboard_views.index, name='index'),
    path('dashboard/', include('dashboard.urls')),
    path('accounts/', include('users.urls')),
    path('doctor/', include('doctor.urls')),
    path('login/', auth_views.LoginView.as_view(template_name='users/login.html'), name='login'),
    path('logout/', auth_views.LogoutView.as_view(template_name='users/logout.html'), name='logout'),
    path('password-reset/', auth_views.PasswordResetView.as_view(template_name='users/password_reset.html'), name='password_reset'),
    path('password-reset/done/', auth_views.PasswordResetDoneView.as_view(template_name='users/password_reset_done.html'),
        name='password_reset_done'),
    path('password-reset-confirm/<uidb64>/<token>/',
        auth_views.PasswordResetConfirmView.as_view(template_name='users/password_reset_confirm.html'), name='password_reset_confirm'),
    path('password-reset-complete/',
        auth_views.PasswordResetCompleteView.as_view(template_name='users/password_reset_complete.html'),
        name='password_reset_complete'),
    path('pharmacist/', include('pharmacist.urls')),
    path('api/', include('api.urls'))
]
```

Smartblister settings.py

```
"""
Django settings for smartblister project.

Generated by 'django-admin startproject' using Django 3.0.7.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/
"""
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
import os
import json

with open('/etc/config.json') as config_file:
    config = json.load(config_file)

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
TEMPLATE_DIR = os.path.join(BASE_DIR, 'templates')
STATIC_DIR = os.path.join(BASE_DIR, 'static')
MEDIA_DIR = os.path.join(BASE_DIR, 'media')

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
#SECRET_KEY = os.environ.get('DJANGO_SECRET_KEY')
SECRET_KEY = config['SECRET_KEY']

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = False

ALLOWED_HOSTS = []
ALLOWED_HOSTS.append(config.get('HOST'))

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'dashboard',
    'users',
    'doctor',
    'pharmacist',
    'api',
    'crispy_forms',
    'rest_framework',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django_session_timeout.middleware.SessionTimeoutMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'smartblister.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'smartblister.wsgi.application'
```

```
# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'el-GR'

TIME_ZONE = 'Europe/Athens'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/
#STATIC_ROOT = STATIC_DIR

STATIC_URL = '/static/'

STATICFILES_DIRS = [STATIC_DIR, ]

MEDIA_ROOT = MEDIA_DIR

MEDIA_URL = '/media/'

CRISPY_TEMPLATE_PACK = 'bootstrap4'

LOGIN_REDIRECT_URL = 'dashboard:index'

LOGIN_URL = 'login'

EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

EMAIL_HOST = 'smtp.gmail.com'

EMAIL_PORT = 587

EMAIL_USE_TLS = True

#EMAIL_HOST_USER = os.environ.get('EMAIL_USER')
EMAIL_HOST_USER = config.get('EMAIL_USER')

#EMAIL_HOST_PASSWORD = os.environ.get('EMAIL_PASS')
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
EMAIL_HOST_PASSWORD = config.get('EMAIL_PASS')
```

```
SESSION_EXPIRE_SECONDS = 300 # 300 seconds = 5 minutes
```

```
SESSION_EXPIRE_AFTER_LAST_ACTIVITY = True
```

```
SESSION_EXPIRE_AT_BROWSER_CLOSE = True
```

Doctor models.py

```
from django.db import models
from users.models import PatientProfile, DoctorProfile
from django.utils import timezone
from django.contrib.auth.models import User
```

```
class MonitoringRequest(models.Model):
    doctor = models.ForeignKey(DoctorProfile, on_delete=models.SET_NULL, blank=True, null=True)
    patient = models.ForeignKey(PatientProfile, on_delete=models.SET_NULL, blank=True, null=True)
    date_requested = models.DateField(default=timezone.now)
    token = models.CharField(max_length=100)
    date_accepted = models.DateField(blank=True, null=True)

    def __str__(self):
        return self.doctor.user.last_name + ' ' + self.doctor.user.first_name[:1] + ' -> ' + self.patient.user.last_name + ' ' + self.patient.user.first_name[:1]
```

```
class Frequency(models.Model):
    description = models.CharField(max_length=50)

    def __str__(self):
        return self.description
```

```
class Duration(models.Model):
    description = models.CharField(max_length=50)

    def __str__(self):
        return self.description
```

```
class Prescription(models.Model):
    quantity_choices = (
        (1,'1'),
        (2,'2'),
        (3,'3')
    )

    frequency_choices = (
        (1,'1 φορά την ημέρα'),
        (2,'2 φορές την ημέρα'),
        (3,'3 φορές την ημέρα'),
        (4,'4 φορές την ημέρα'),
        (5,'1 φορά την εβδομάδα'),
        (6,'2 φορές την εβδομάδα'),
        (7,'3 φορές την εβδομάδα')
    )

    duration_choices = (
        (1,'για 7 ημέρες'),
        (2,'για 10 ημέρες'),
        (3,'για 2 εβδομάδες'),
        (4,'για 1 μήνα'),
        (5,'για 2 μήνες'),
        (6,'για 3 μήνες'),
        (7,'για 6 μήνες'),
        (8,'για ένα έτος')
    )

    patient = models.ForeignKey(PatientProfile, null = True, blank=True, on_delete=models.PROTECT)
    doctor = models.ForeignKey(DoctorProfile, null = True, blank=True, on_delete=models.PROTECT)
    date_issued = models.DateField(default=timezone.now)
    illness = models.CharField(max_length=255)
    medicine = models.CharField(max_length=255)
    quantity = models.IntegerField(choices=quantity_choices,default=1)
    dosage = models.IntegerField(choices=quantity_choices,default=1)
    frequency = models.ForeignKey(Frequency, on_delete=models.PROTECT, default=1)
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
duration = models.ForeignKey(Duration, on_delete=models.PROTECT, default=1)
#duration = models.IntegerField(choices=duration_choices,default=1)

def __str__(self):
    return self.medicine+' '+str(self.date_issued)

"""class BlisterAction(models.Model):
    blisterPrescription = models.ForeignKey(BlisterPrescription, on_delete=models.PROTECT)
    date_removed = models.DateTimeField(default=timezone.now)

    def __str__(self):
        return self.blisterPrescription.blister.serial+' '+str(self.date_removed)"""
```

Doctor view.py

```
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from .forms import MonitoringRequestForm, PatientSearchForm, PrescriptionAddForm
from django.core.mail import send_mail
from users.models import PatientProfile, DoctorProfile, UserProfile
from doctor.models import MonitoringRequest, Prescription
from pharmacist.models import BlisterAction, BlisterPrescription
from django.contrib.auth.models import User
from django.utils import timezone
from datetime import timedelta
from django.contrib import messages
from django.shortcuts import get_object_or_404

@login_required
def monitoring_request(request):
    if request.method == 'POST':
        form = PatientSearchForm(request.POST)
        if form.is_valid():
            amka = form.cleaned_data['amka']
            users = UserProfile.objects.filter(amka__startswith=amka)
            users = users.filter(is_patient=True)
        else:
            users = None

    else:
        form = PatientSearchForm()
        users = None

    context = {
        'title': 'Αίτημα πρόσβασης smartblister',
        'form': form,
        'users': users
    }
    return render(request, 'doctor/monitoring_request.html', context)

@login_required
def monitoring_request_send(request):
    if request.method == 'POST':
        import json, hashlib

        with open('/etc/config.json') as config_file:
            config = json.load(config_file)

        EMAIL_HOST_USER = config.get('EMAIL_USER')
        patientmainprofile = UserProfile.objects.get(id=request.POST['patient'])
        try:
            patient = PatientProfile.objects.get(user=patientmainprofile.user)
            doctor = DoctorProfile.objects.get(user=request.user.id)
            date_requested = timezone.now()
            pre_token = str(doctor.id)+str(patient.id)+str(date_requested)
            token = hashlib.sha256(pre_token.encode('utf-8')).hexdigest()
            req =
MonitoringRequest.objects.create(doctor=doctor,patient=patient,date_requested=date_requested,token=token)
            req.save()
            host = config.get('HOST_URL')
            subject = 'Smartblister - Αίτηση πρόσβασης στα στοιχεία του smartblister'
            sender = EMAIL_HOST_USER
            recipient = patient.user.email
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
plain_message = 'Ο ιατρός '+ request.user.last_name + ' '+ request.user.first_name + ' με ειδικότητα ' +
doctor.speciality + ' έχει αιτηθεί πρόσβαση στα στοιχεία του smartblister σας. Αν Θέλετε να κάνετε αποδοχή της αίτησης κάντε κλικ στον
παρακάτω σύνδεσμο; '+host+'smartblister/doctor/monitoring_accept/'+token
html_message = '<p>Ο ιατρός <b>'+ request.user.last_name + ' '+ request.user.first_name + '</b> με
ειδικότητα <b>'+ doctor.speciality + '</b> έχει αιτηθεί πρόσβαση στα στοιχεία του smartblister σας. Αν Θέλετε να κάνετε αποδοχή της
αίτησης κάντε κλικ στον παρακάτω σύνδεσμο;</p><p><a
href="'+host+"smartblister/doctor/monitoring_accept/'+token+'"/>'+host+"smartblister/doctor/monitoring_accept/'+token+'</a></p>'
send_mail(subject,plain_message, sender, [recipient,], fail_silently=False,html_message=html_message)
messages.success(request,f'Έχει αποσταλεί στον ασθενή με όνομα {patient.user.last_name}
{patient.user.first_name} αίτημα μέσω ηλεκτρονικού μηνύματος για εξουσιοδότηση πρόσβασης στα στοιχεία του smartblister που διαθέτει.
Θα έχετε πρόσβαση στα σχετικά στοιχεία του ασθενούς όταν γίνει αποδοχή του αιτήματός σας. Αμέσως μετά την αποδοχή του αιτήματος θα
λάβετε την σχετική ενημέρωση μέσω ηλεκτρονικού μηνύματος.')
except PatientProfile.DoesNotExist:
    messages.warning(request,f'Δεν υπάρχει καταχωρημένος ασθενής με το στοιχεία που δηλώσατε')

else:
    messages.warning(request,f'Θσ πρέπει να συνδεθείτε ως ιατρός και να αποστείλετε αίτημα πρόσβασης σε στοιχεία
smartblister του ασθενούς που θα έχετε πρώτα αναζητήσει.')

return render(request,'dashboard/index.html')

def monitoring_accept(request,token):
    req = MonitoringRequest.objects.filter(token=token).first()
    if req:
        doctor = req.doctor
        patient = req.patient
        patient.doctor = doctor
        patient.save()
        req.date_accepted = timezone.now()
        req.save()
        import json

        with open('/etc/config.json') as config_file:
            config = json.load(config_file)

        EMAIL_HOST_USER = config.get('EMAIL_USER')
        subject = 'Smartblister - Αποδοχή αιτήματος πρόσβασης στα στοιχεία του smartblister'
        sender = EMAIL_HOST_USER
        recipient = doctor.user.email
        plain_message = 'Ο ασθενής '+patient.user.last_name+' '+patient.user.first_name+' έχει αποδεχθεί το αίτημά σας για
πρόσβαση στα στοιχεία του smartblister που διαθέτει'
        html_message = 'Ο ασθενής <b>'+patient.user.last_name+' '+patient.user.first_name+'</b> έχει αποδεχθεί το αίτημά
σας για πρόσβαση στα στοιχεία του smartblister που διαθέτει.'
        send_mail(subject,plain_message, sender, [recipient,], fail_silently=False,html_message=html_message)
        messages.success(request,f'Έχετε αποδεχθεί το αίτημα πρόσβασης του ιατρού με όνομα {doctor.user.last_name}
{doctor.user.first_name} στα στοιχεία του smartblister που διαθέτετε. Μπορείτε οποιαδήποτε στιγμή να ανατρέξετε την ενέργεια αυτή
πηγαίνοντας στο προφίλ σας αφού πρώτα συνδεθείτε στο σύστημα.')
    else:
        messages.warning(request,f'Τα στοιχεία δεν είναι έγκυρα.')

    context = {
        'title': 'Αίτημα πρόσβασης smartblister',
        'role': 'patient',
    }
    return render(request,'dashboard/index.html',context)

@login_required
def prescription(request):
    access_rights = True
    if request.method == 'POST':
        form = PatientSearchForm(request.POST)
        form1 = PrescriptionAddForm(request.POST)
        if form.is_valid():
            amka = form.cleaned_data['amka']
            try:
                patient = PatientProfile.objects.get(user__userprofile__amka=amka)
                prescriptions = patient.prescription_set.all().order_by('-date_issued')
                if patient.doctor is None:
                    access_rights = False
            except PatientProfile.DoesNotExist:
                patient = None
                prescriptions = None
        else:
            patient = None
            prescription = None
    else:
        patient = None
        prescription = None
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
form = PatientSearchForm()
form1 = PrescriptionAddForm()
patient = None
prescriptions = None

context = {
    'title': 'Συνταγογράφηση',
    'form': form,
    'form1': form1,
    'patient': patient,
    'prescriptions': prescriptions,
    'access_rights': access_rights
}

return render(request, 'doctor/prescription.html', context)

@login_required
def actions(request):
    if request.method == 'POST':
        form = PatientSearchForm(request.POST)
        form1 = BlisterAddForm(request.POST)
        if form.is_valid():
            amka = form.cleaned_data['amka']
            try:
                patient = PatientProfile.objects.get(user__userprofile__amka=amka)
                blisters = patient.blister_set.all().order_by('-charge_date')
            except PatientProfile.DoesNotExist:
                patient = None
                blisters = None
        else:
            patient = None
            blisters = None
    else:
        form = PatientSearchForm()
        form1 = BlisterAddForm()
        patient = None
        blisters = None

    context = {
        'title': 'Χρέωση smartblister',
        'form': form,
        'form1': form1,
        'patient': patient,
        'blisters': blisters
    }

    return render(request, 'doctor/prescription_list.html', context)

@login_required
def actions(request, prescription_id):
    try:
        actions = BlisterAction.objects.filter(blisterPrescription__prescription__id=prescription_id).order_by('-date_removed')
        prescription = Prescription.objects.get(id=prescription_id)
        patient = prescription.patient
    except BlisterPrescription.DoesNotExist:
        actions = None
        prescription = None
        patient = None

    context = {
        'title': 'Φαρμακευτική Συνέπεια',
        'actions': actions,
        'prescription': prescription,
        'patient': patient
    }

    return render(request, 'doctor/action_list.html', context)

@login_required
def chargedPrescriptions(request, patient_id):
    access_rights = True
    if request.method == 'POST':
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
form = PatientSearchForm(request.POST)
form1 = PrescriptionAddForm(request.POST)
if form.is_valid():
    amka = form.cleaned_data['amka']
    try:
        patient = PatientProfile.objects.get(user__userprofile__amka=amka)
        prescriptions = patient.prescription_set.all().order_by('-date_issued')
    except PatientProfile.DoesNotExist:
        patient = None
        prescriptions = None
    if patient.doctor is None:
        access_rights = False
else:
    patient = None
    prescriptions = None
else:
    form = PatientSearchForm()
    form1 = PrescriptionAddForm()
    try:
        patient = PatientProfile.objects.get(id=patient_id)
        prescriptions = patient.prescription_set.all().order_by('-date_issued')
    except PatientProfile.DoesNotExist:
        patient = None
        blisters = None
    if patient.doctor is None:
        access_rights = False

context = {
    'title': 'Συνταγογράφηση',
    'form': form,
    'form1': form1,
    'patient': patient,
    'prescriptions': prescriptions,
    'access_rights': access_rights
}

return render(request, 'doctor/prescription.html', context)

@login_required
def deletePrescription(request, prescription_id):
    try:
        prescription = Prescription.objects.get(id=prescription_id)
        patient = prescription.patient
        prescription.delete()
        return redirect('doctor:chargedPrescriptions', patient_id=patient.id)
    except Prescription.DoesNotExist:
        messages.warning(request, f'Δεν υπάρχει συνταγή με τα στοιχεία που δώσατε')
        return redirect('doctor:prescription')

@login_required
def addPrescription(request, patient_id):
    access_rights = True
    if request.method == 'POST':
        form = PrescriptionAddForm(request.POST)
        patient = PatientProfile.objects.get(id=patient_id)
        if form.is_valid():
            print("valid form")
            prescription = form.save(commit=False)
            prescription.patient = patient
            prescription.doctor = request.user.doctorprofile
            prescription.save()
            prescriptions = patient.prescription_set.all().order_by('-date_issued')
            #messages.success(request, f'Το blister αποθηκεύτηκε')
        else:
            print("not valid form")
            print(form.errors)
        if patient.doctor is None:
            access_rights = False
    else:
        print("not posted")
        form = PrescriptionAddForm()
        patient = PatientProfile.objects.get(id=patient_id)
        prescriptions = patient.prescription_set.all()
```



```

context = {
  'title': 'Συνταγογράφηση Ασθενούς',
  'form': form,
  'patient': patient,
  'prescriptions': prescriptions,
  'access_rights': access_rights
}

return redirect('doctor:chargedPrescriptions', patient_id=patient.id)

```

Doctor action_list.html

```

{% extends "dashboard/base.html" %}
{% load crispy_forms_tags %}
{% block body_block %}
<script>
$(document).ready(function(){
//var endpoint = '/api/actions/{{prescription.id}}/'
var endpoint = "/smartblister/api/actions/{{prescription.id}}/"
var rest_data = []
var rest_labels = []
$.ajax({
  method: "GET",
  url: endpoint,
  success: function(data){
    for (i=0;i<data.length;i++){
      rest_data.push(data[i].removals)
      rest_labels.push(data[i].date_removed__date)
    }
    var ctx = document.getElementById('myChart');
    var myChart = new Chart(ctx, {
      type: 'bar',
      data: {
        labels: rest_labels,
        datasets: [{
          label: 'Ημερήσια Κατανάλωση Δισκίων',
          data: rest_data,
          borderWidth: 1
        }]
      },
      options: {
        scales: {
          yAxes: [{
            ticks: {
              beginAtZero: true
            }
          }]
        }
      }
    });
  },
  error: function(error_data){
    console.log("error")
    console.log(error_data)
  }
})
</script>

<div class="content-section">
<div class="container mb-4">
  <legend class="mb-4">Φαρμακευτική Συνέπεια Ασθενούς - {{patient.user.last_name}} {{patient.user.first_name}}</legend>
</div>
<div class="table-responsive">
<table class="table">
<thead class="thead-light">
<tr>
<!--<th>Α/Α</th-->
<th>Πάθηση</th>
<th>Φάρμακο</th>
<th>Ποσότητα</th>
<th>Δόση<br>σε Δισκία</th>

```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
<th>Συχνότητα</th>
<th>Διάρκεια</th>
<th>Ημ/ρία<br>Συνταγογράφησης</th>
</tr>
</thead>
<tbody>
<tr>
<!--<td>{{ forloop.counter }}</td>-->
<td>{{ prescription.illness }}</td>
<td>{{ prescription.medicine }}</td>
<td>{{ prescription.quantity }}</td>
<td>{{ prescription.dosage }}</td>
<td>{{ prescription.frequency.description }}</td>
<td>{{ prescription.duration.description }}</td>
<td>{{ prescription.date_issued }}</td>
</tr>
</tbody>
</table>
</div>
<canvas id="myChart"></canvas>
<div class="text-center mt-4"><h5>Αναλυτικές Κινήσεις στο Blister</h5></div>
<div class="container">
<div class="table-responsive">
<table class="table">
<thead>
<tr>
<th>Α/Α</th>
<th>Ημ/ρία Αφαίρεσης</th>
</tr>
</thead>
<tbody>
{% if actions %}
{% for action in actions %}
<tr>
<td>{{ forloop.counter }}</td>
<td>{{ action.date_removed }}</td>
</tr>
{% endfor %}
{% else %}
<tr>
<td>Δεν υπάρχουν δεδομένα</td>
<td></td>
</tr>
{% endif %}
</tbody>
</table>
</div>
</div>
<div class="d-flex flex-row">
<a href="{% url 'doctor:chargedPrescriptions' patient.id %}" class="btn btn-outline-info">Επιστροφή</a>
</div>
</div>
{% endblock %}
```

API view.py

```
from django.shortcuts import render
from django.http import JsonResponse
from django.db.models import Count
from rest_framework.decorators import api_view
from rest_framework.response import Response
from rest_framework import status
from rest_framework import serializers
from .serializers import TaskSerializer, ChargeSerializer, BlisterSerializer, BlisterPrescriptionSerializer, BlisterActionSerializer,
RemovalSerializer
from .models import Task
from pharmacist.models import Blister, BlisterPrescription, BlisterAction

@api_view(['GET'])
def apiOverview(request):
    #return JsonResponse("API BASE POINT", safe=False)
    api_urls = {
        # 'List': '/task-list/',
        # 'Detail View': '/task-detail/<str:pk>/',
        # 'Create': '/task-create/',
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
#'Update': '/task-update/<str:pk>/',
#'Delete': '/task-delete/<str:pk>/',
'Service': '/service/',
'Status': '/status/<str:serial>/',
'PillRemoved': '/pillRemoved/<str:serial>/',
'Actions': '/actions/<int:prescription_id>/',
'DetailedActions': '/detailedActions/<int:prescription_id>/'
}

#return JsonResponse("API BASE POINT", safe=False)
return Response(api_urls)

@api_view(['GET'])
def service(request):
    return Response("Smartblister Rest Service is Up and Running", status=200)

@api_view(['GET'])
def status(request, pk):
    try:
        blister = Blister.objects.get(serial=pk)
        blisterPrescription = BlisterPrescription.objects.filter(blister=blister).first()
        serializer = BlisterPrescriptionSerializer(blisterPrescription, many=False)
        return Response(serializer.data, status=200)
    except blister.DoesNotExist:
        return Response("Not found", status=404)
    except blisterPrescription.DoesNotExist:
        return Response("Not found", status=404)

@api_view(['POST'])
def pillRemoved(request, serial):
    blister = Blister.objects.get(serial=serial)
    #blisterPrescription = BlisterPrescription.objects.get(blister=blister)
    blisterPrescription = BlisterPrescription.objects.filter(blister=blister).first()
    p = BlisterAction(blisterPrescription=blisterPrescription)
    p.save()
    #serializer1 = BlisterPrescriptionSerializer(blisterPrescription, many=False)
    #serializer = BlisterActionSerializer(data=serializer1)

    #if serializer.is_valid():
    #    print("valid data")
    #    serializer.save()
    #else:
    #    print("not valid")
    #    print(serializer)

    #return Response(serializer.data, status=201)
    return Response("Η αφαίρεση δισκίου καταχωρήθηκε", status=200)

@api_view(['GET'])
def actions(request, prescription_id):
    data =
    list(BlisterAction.objects.filter(blisterPrescription__prescription_id=prescription_id).values('date_removed__date').annotate(removals=Count('date_removed__date')).order_by('date_removed__date'))
    #actions =
    BlisterAction.objects.filter(blisterPrescription__prescription_id=prescription_id).values('date_removed__date').annotate(removals=Count('date_removed__date')).order_by('date_removed__date')

    #serializer = RemovalSerializer(actions, many=True)
    return JsonResponse(data, safe=False, status=200)
    #return Response(serializer.data, status=200)

@api_view(['GET'])
def detailedActions(request, prescription_id):
    actions = BlisterAction.objects.filter(blisterPrescription__prescription_id=prescription_id).order_by('-date_removed')
    serializer = BlisterActionSerializer(actions, many=True)
    return Response(serializer.data, status=200)

@api_view(['GET'])
def taskList(request):
    tasks = Task.objects.all()
    serializer = TaskSerializer(tasks, many=True)
    return Response(serializer.data)

@api_view(['GET'])
def taskDetail(request, pk):
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
tasks = Task.objects.get(id=pk)
serializer = TaskSerializer(tasks, many=False)
return Response(serializer.data)

@api_view(['POST'])
def taskCreate(request):
    serializer = TaskSerializer(data=request.data)

    if serializer.is_valid():
        serializer.save()

    return Response(serializer.data, status=status.HTTP_201_CREATED)

@api_view(['POST'])
def taskUpdate(request, pk):
    try:
        task = Task.objects.get(id=pk)
        serializer = TaskSerializer(instance=task, data=request.data)

        if serializer.is_valid():
            serializer.save()

        return Response(serializer.data)
    except Task.DoesNotExist:
        return Response("Not found - Nothing to Update", status=404)

@api_view(['DELETE'])
def taskDelete(request, pk):
    task = Task.objects.get(id=pk)
    serializer = TaskSerializer(task, many=False)
    task.delete()

    #return Response('Item successfully delete!')
    return Response(serializer.data, status=status.HTTP_200_OK)
```

API serializers.py

```
from rest_framework import serializers
from .models import Task
from doctor.models import Prescription
from pharmacist.models import Blister, BlisterPrescription, BlisterAction
from users.models import PatientProfile, DoctorProfile
from django.contrib.auth.models import User

class TaskSerializer(serializers.ModelSerializer):
    class Meta:
        model = Task
        fields = '__all__'

class ChargeSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blister
        fields = '__all__'

class UserSerializer(serializers.ModelSerializer):
    class Meta:
        model = User
        fields = ['last_name', 'first_name']

class DoctorSerializer(serializers.ModelSerializer):
    user = UserSerializer(many=False)
    class Meta:
        model = DoctorProfile
        fields = '__all__'

class PatientSerializer(serializers.ModelSerializer):
    user = UserSerializer(many=False)
    doctor = DoctorSerializer(many=False)
    class Meta:
        model = PatientProfile
        fields = ['user', 'doctor']

class PrescriptionSerializer(serializers.ModelSerializer):
    doctor = DoctorSerializer(many=False)
    class Meta:
        model = Prescription
```

Παράρτημα - Ενδεικτικός Κώδικας Εφαρμογής

```
fields = ['doctor','illness','medicine']

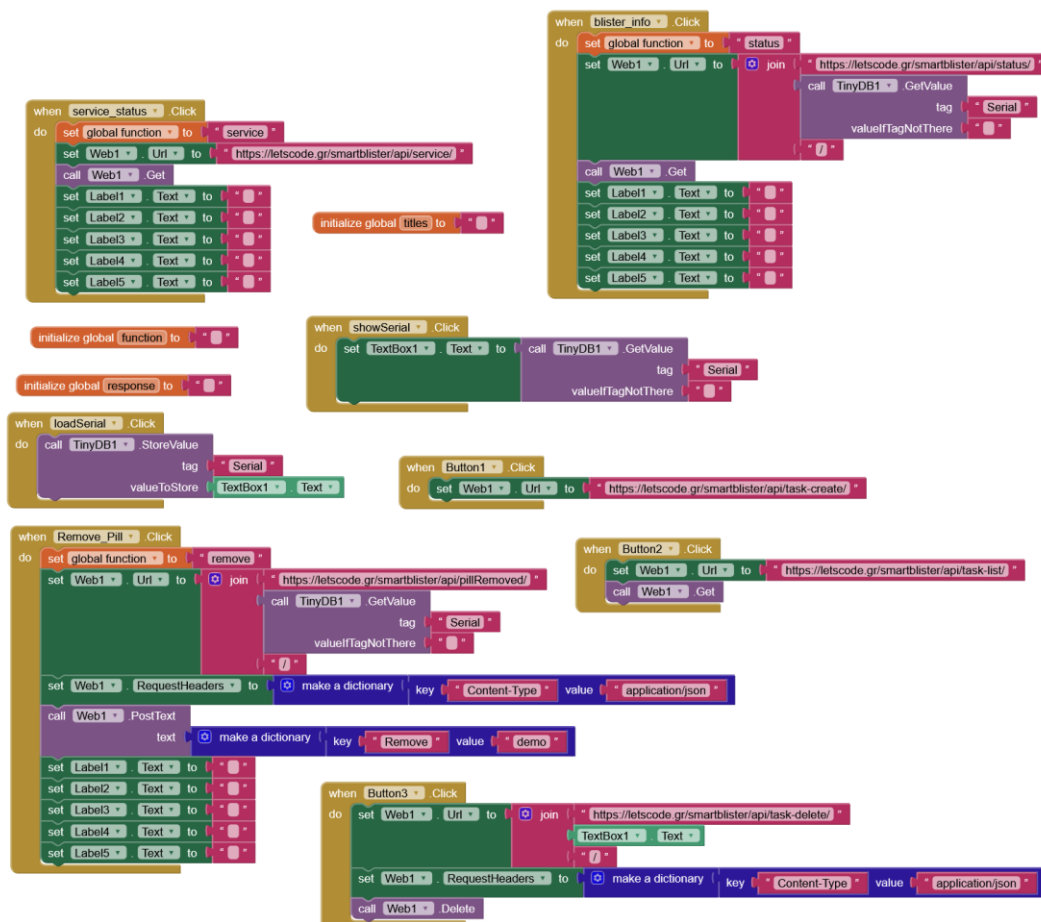
class BlisterSerializer(serializers.ModelSerializer):
    patient = PatientSerializer(many=False)
    class Meta:
        model = Blister
        fields = ['serial','charge_date','patient']

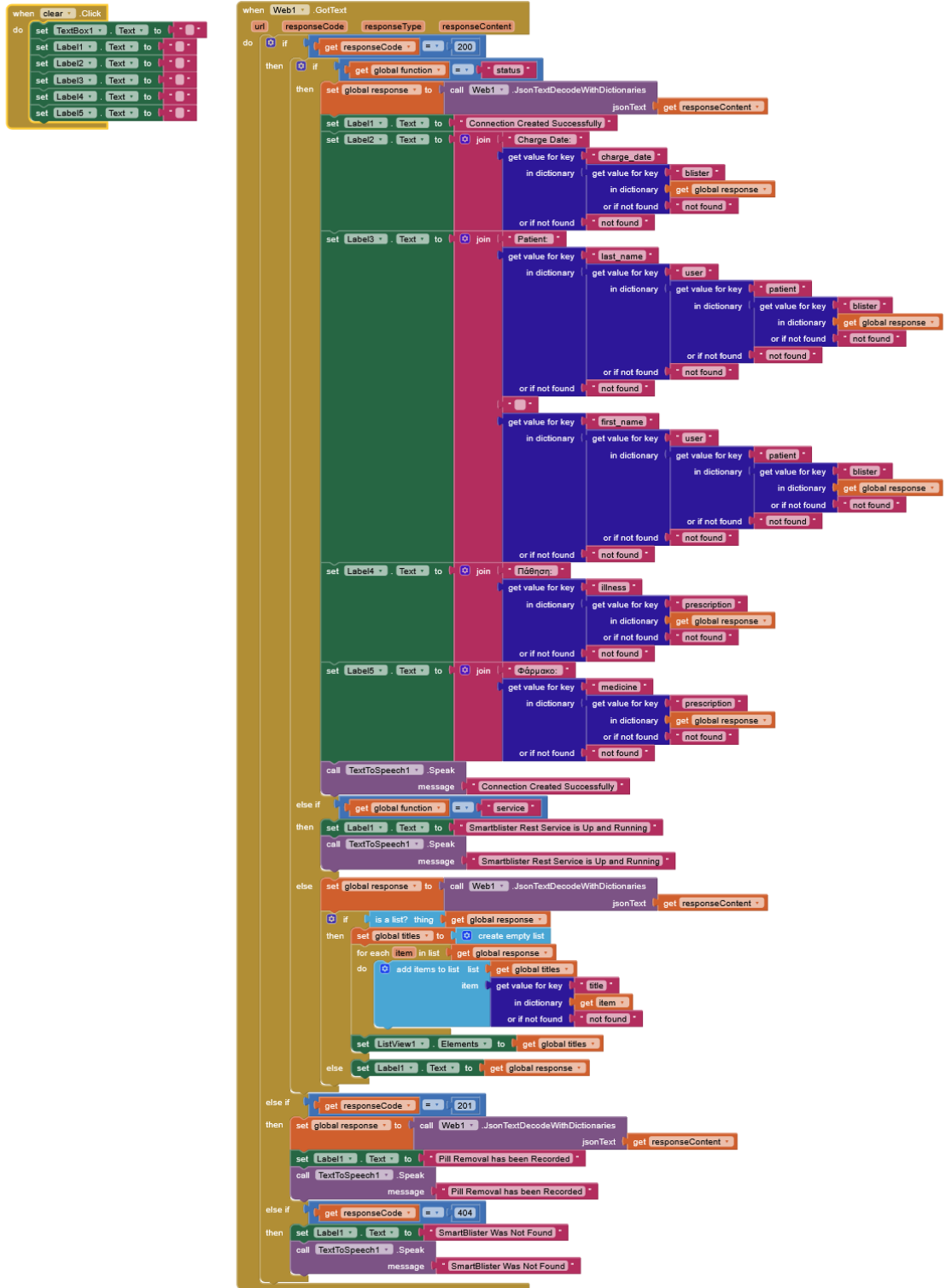
class BlisterPrescriptionSerializer(serializers.ModelSerializer):
    blister = BlisterSerializer(many=False)
    prescription = PrescriptionSerializer(many=False)
    class Meta:
        model = BlisterPrescription
        fields = ['blister','prescription']

class BlisterActionSerializer(serializers.ModelSerializer):
    class Meta:
        model = BlisterAction
        fields = ['date_removed']

class RemovalSerializer(serializers.Serializer):
    date_removed__date = serializers.DateField()
    removals = serializers.IntegerField()
```

Αντίστοιχα παρακάτω παραθέτω τον κώδικα της mobile εφαρμογής η οποία έχει υλοποιηθεί στο γραφικό περιβάλλον του MIT APP inventor.





Σχήμα 43. Γραφικός κώδικας mobile εφαρμογής σε περιβάλλον MIT APP inventor

Βιβλιογραφία

- [1] World Health Organization, "Adherence to long-term therapies : evidence for action.," <https://iris.who.int/handle/10665/42682>, 2003.
- [2] M. T. B. MD and Jennifer K. Bussell MD, "Medication Adherence: WHO Cares?," *Mayo Clinic Proceedings* <https://doi.org/10.4065/mcp.2010.0575>, vol. 86, no. 4, 2011.
- [3] «Συμμόρφωση,» *ADHERENCE-NCDS* <https://adherence-ncds.ipoke.gr/>.
- [4] Shelly A Vik, Colleen J Maxwell and David B Hogan, "Measurement, Correlates, and Health Outcomes of Medication Adherence Among Seniors," *Sage Journal* <https://doi.org/10.1345/aph.1D252>, vol. 38, no. 2, 2004.
- [5] Dawn I. Velligan, Mei Wang, Pamela Diamond, David C. Glahn, Desiree Castillo, Scott Bendle, Y. W. Francis Lam, Larry Ereshefsky and Alexander L. Miller, "Relationships Among Subjective and Objective Measures of Adherence to Oral Antipsychotic Medications," <https://doi.org/10.1176/ps.2007.58.9.1187>, vol. 58, no. 9, 2007.
- [6] Mary Jayne Johnson, Mary Williams and Elaine S. Marshall, "Adherent and Nonadherent Medication-Taking in Elderly Hypertensive Patients," *Clinical Nursing Research* <https://doi.org/10.1177/10547739922158331>, vol. 8, no. 4, 1999.
- [7] Susan E. Andrade, Kristijan H. Kahler, Feride Frech και K. Arnold Chan, «Methods for evaluation of medication adherence and persistence using automated databases,» *Pharmacoepidemiology and Drug Safety* <https://doi.org/10.1002/pds.1230>, τόμ. 15, αρ. 8, 2006.
- [8] Bernard Vrijens, Sabina De Geest, Dyfrig A. Hughes, Kardas Przemyslaw, Jenny Demonceau, Todd Ruppert, Fabienne Dobbels, Emily Fargher, Valerie Morrison, Pawel Lewek, Michal Matyjaszczyk, Comfort Mshelia, Wendy Clyne, Jeffrey K. Aronson and J. Urquhart, "A new taxonomy for describing and defining adherence to medications," *British Journal of Clinical Pharmacology* <https://doi.org/10.1111/j.1365-2125.2012.04167.x>, vol. 73, no. 5, 2012.
- [9] Solomon Zannos, Dimitris Nikitopoulos, Vassilis Gryparis, Lazaros Tzounis, Konstantina Nikita, Ilias Maglogiannis and Panayiotis Tsanakas, "Promoting Medication Adherence by Redesigning Medical Blisters - The Smartblister approach," *RESEARCH - CREATE - INNOVATE (project code: T1EDK- 02506)*.
- [10] D. C. Marinescu, «Cloud Computing Theory and Practice,» <https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/212-Cloud-Computing-Theory-and-Practice-Dan-C.-Marinescu-Edisi-2-2017.pdf>, 2018.
- [11] Peter Mell and Timothy Grance, "The NIST Definition of Cloud Computing," *NIST* <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>, 2011.
- [12] Q. Zhang, Lu Cheng και Raouf Boutaba, «Cloud computing: state-of-the-art and research challenges,» *Journal of Internet Services and Applications* <https://jisajournal.springeropen.com/articles/10.1007/s13174-010-0007-6>, 2010.
- [13] R. A. Mouha, «Internet of Things (IoT),» *Scientific Research Publishing* <https://doi.org/10.4236/jdaip.2021.92006>, 2021.
- [14] «Remote patient monitoring,» *Wikipedia* https://en.wikipedia.org/wiki/Remote_patient_monitoring, 2020.

- [15] R. Buyya and Amir Vahid Dastjerdi, "Internet of Things Principles and Paradigms," *Morgan Kaufmann*
https://dphoto.lecturer.pens.ac.id/lecture_notes/internet_of_things/Internet%20of%20Things%20Principles%20and%20Paradigms.pdf, 2016.
- [16] «Εξ αποστάσεως δυνατότητες: Τηλεϊατρική διάγνωση στην κροατική ύπαιθρο,»
https://next-generation-eu.europa.eu/remote-possibilities-telemedical-diagnostics-rural-croatia-2024-02-02_el, 2024.